WEB SCRAPING (IMDB) USING PYTHON

Article · January 2022

CITATIONS
15

READS
1,887

4 authors, including:

Dr.B.Narendra Kumar Rao
Mohan Babu University
41 PUBLICATIONS **103** CITATIONS
SEE PROFILE

Beebi Naseeba
VIT University Amaravati
51 PUBLICATIONS **96** CITATIONS
SEE PROFILE

Nagendra Panini Challa
VIT-AP University Amaravati
80 PUBLICATIONS **257** CITATIONS
SEE PROFILE

TELEMATIQUE
ISSN: 1856-4194
Volume 21 Issue 1, 2022
235-247

WEB SCRAPING (IMDB) USING PYTHON

Narendra Kumar Rao[1], Beebi Naseeba[2], Nagendra Panini Challa[3], S Chakrvarthi[4]

[1,4] Department of Computer Science and Engineering, Sri Vidhyanikethan Engineering College, Tirupathi, India

[2,3] School of Computer Science and Engineering, VIT-AP University, Amaravati

Abstract:

Background: Web scraping is process of obtaining text information from web pages. Most of the analysis focusing web scraping is about automated web data extraction. In process of the web data extraction, we first create a DOM tree and then extract the relevant data through this tree. The construction technique of DOM tree boosts the time cost depending on the design structure of the DOM Tree. In this paper we accurately predict particular genre that is popular and well-versed in a particular year based on the analysis of data in real-time. For the latter, we are going to use IMDB website as a source and use python as a intermediary language to pre-process them and clean the data using built in libraries such as BeautifulSoup, HTTPAdaptor, Seaborn etc., and by using subtle python packages such as Numpy, Pandas ,MatPlotLib etc.., On addition to this we have also incorporated an API add-on: in either ways, An API is the carrier which hands over your requisition to the provider and then acknowledges the response to the user. Having all these done we will have various type of visualizations over different parameters and also based on requirement we add certain parameters to facilitate our need for typical analysis of provided data by providing the scores for each genre in each individual year.

Keywords: BeautifulSoup, Seaborn, Numpy, Panda, JSON

1. Introduction

Over a decade, the number of data in social networking sites has increased drastically. There is so much content out there today, that only a good import filter is essential if you are going to save time. Take movies as an example. If we had a movie rating tool before it was released, we could save a lot of time by not watching movies with lower ratings. We do not need to re-emphasize the importance of the time and money we have in our lives today. We are creating a movie rating forecast, using data from the IMDB website.

Data Analysis is the process of evaluating and completely comprehending a set of data in order to find answers to questions. The analysis process includes identifying problems, addressing the availability of required data, understanding which method can assist in solving the exciting challenge, and presenting the results. For this aim, the data is incorporated into a series of phases, beginning with its specification and progressing through assembling, organizing, cleaning, applying algorithms, and obtaining the final output. Scraping data from the web and announcing it through visuals are two excellent methods for automatically creating content on the internet. A web scraper is an Application Programming interface which implements to extract relevant data from a website. Web scraping mechanisms, organizations, and freely open data are all available for free thanks to corporations like Amazon AWS and Google. Web scraping is an important technique because it enables the extraction of material in the form of news from a variety of sources quickly and efficiently.

The scraping process normally follows these similar steps:

- Open the website that you want to scrap, inspect it to find the HTLM tag of the element you want.
- If you need to automate some tasks (click somewhere, fill out a search field, etc…), use Selenium.
- Then store the webpage into an object and extract the desired element with its HTML tag.
- If you need multiple items, you can run a loop to store them into a list, and create a dataframe base on that.

The main is to develop an API that facilitates the user to incorporate various information, from multiple websites, pre-process them and by drawing visualizations to accurately comprehend the data and provide business insights across multiple disciplines. Web scraping is limited to coded and pre-configured layouts on a partner website, in the event that a new layout is found, the web script will fail to `return the desired output. The API service depends on the partner's website and in some cases is limited by it, in the event that the limit is reached, we will not be able to use this API service for a while until the restrictions are reset. Python scripts for the application will run on a server-free basis, performance and reliability depend on availability of these services and resources.

The fundamental goal is to extract data from various sources using a web crawler software written in Python 3.6.This application will be handy when needed to comprehend any data related to films and also being open source, it can perform tasks as per needs of the user. We can also extract data from other commercial websites as well but those doesn't provide us as this API, enables to analyse comments, ratings, or any attribute pertaining to any movie .The application should successfully work for any specific websites with or without static web pages, on any chosen website.

.

2. Literature Survey

Understanding how the data extraction process has grown is crucial since the techniques used in this important online scraping method of scraping have existed for as long as the internet. The result of business web scraping has been easy company profits and the integration of things like weakening a competitor's unique value, tracking, and promotional activities, diverting APIs, and internal and external robberies and information.

There are many available in Python to facilitate data processing, data analysis, data recognition, and machine learning functions. These libraries are useful for procuring data on websites. Provides an amicable-to-use and efficient way to set up HTTP requests. Some of the most popular libraries are of:

Table 1. Libraries supporting Web Scraping for Data Analysis

Library	Genre
Numpy	Python based library that supports for many mathematical activities across all major, diverse groups and matriculants.
Pandas	Pandas library is one among most popular and amicable-to-use libraries available in technology field for al contemporaries. Allows easy data manipulation of data erasing and data analysis.
Matplotlib	library provides easy ways to produce vertical or interactive box boxes, scatter pieces, bar charts, and line graphs.

Seaborn	This is alternate data viewing library built on matplotlib that allows for attractive mathematical graphs. Allows you to amicably view intervals of confidence, distribution, and other graph models.
Scipy	Scipy is a computer-assisted science library that provides line algebra, development, and mathematical operations.

In this paper we establish a python script that scrapes the IMDB website by sending request to site and response from it is stored locally in the drive where python file exists and we form a data frame on top of it and then organize a data frame and use it for compelling data visualizations. The Code of Conduct and Integration Procedures for Web Scraper are integrated, explaining how the scraper is organized in advance. Its strategy is split into three different parts: the web scraper pulls the desired links across the web, and then the data is extracted to retrieve data from the source links and finally inserted that data into a csv file. Python language is used for processing.

The extraction process consists of two steps, one at a time to obtain information from the blocks. The first step is learning which HTML tags that refer to the blocks that employ decision tree algorithm. In the second step, the searcher removes the contents of the blocks using the algorithm to match the character unit.

Table 2. Analysis of Papers

S.No	Paper Title	Author name	Findings	Solutions
1	A Novel Web Scraping approach Using the Additional Information Obtained From Web Pages	ERDINÇ UZUN	To determine extraction patterns	Techniques Uses DOM-based and string-based methods. Python, Web Scraping (BeautifulSoup)

				,Implementing Web Scrape
2	Data Analysis by Web Scraping using Python	David Mathew Thomas, Sandeep Mathur	) Examine Data	Python,Web Scraping , (BeautifulSoup) , Implementing Web Scrape
3	Web Scraping Using python	Ryan Mitchell	Web scratching and creeping procedures	Python,Web Scraping , (BeautifulSoup)
4	Web Scraping with Python	Richard Lawson	Successfully scrape data from any website	Python,Web Scraping , (BeautifulSoup)
5	Unstructured Text Documents Summarization with Multi-Stage Clustering	Muhammad yahya saeed, muhammad awais, ramzan talib, Muhammad Younas	To perform improved metadata analysis by relying on pre-extracted text	Corpus processing technique

In this paper we develop a newfangled hypertext detection system as Focused Crawler. The purpose of this concentrated search is to select pages similar to the specific set of topics. Topics are defined not to use keywords, but example texts. Inspite collecting and indexing all Web documents that are accessible to answer any potential ad-hoc queries, the search engine optimizes its crawling margins to find the most relevant links for search, and avoids unnecessary websites and helps ultimately clear the current situation. There are many web crawlers using which we can extract lot of information from multiple websites and visualize the data.

Table 3: Existing Web Crawlers

Library	User Agent	Genre
Googlebot	Googlebot	http://www.google.com/bot.html
Googlebot News	Googlebot-News	Googlebot-News
Googlebot Images	Googlebot Images	Googlebot Images/1.0
Googlebot Videos	Googlebot Videos	Googlebot Videos/1.0
Googlebot Mobile	Googlebot Mobile	http://www.google.com/bot.html
Googlebot Adsense	Googlebot Adsense	Mediapartners-Google

3. Proposed System & Design

The proposed system is based on website you scrape and load it in the JSON format so that you could extract the data on labels collectively. Extraction of structured data is useful for wide range of applications such as data mining, processing information, and archive. Here, in this system we have used to request-response method as well to tell we could arrive at this process using API method and that could be the easiest way possible to run and collect data from any preferred website in matter of sec and also could never think of inappropriate data being extracted.

Steps on how the system will work are:

- Pick the website of desired choice.
- Define the data you want to scrap.
- Write the Session to extract the data using an URL of the website.
- Run the session using http adaptor.
- Review the scraped data.

- Perform visualizations and formulate the decisions.

Fig. 1. Proposed Architecture of Web Scraping API

Algorithm:

Step 1: First the system opens a domain and locates the info about the movie and fetches the first URL's mean the top listing links in the website.

Step 2: Now the system gets the first URL's to crawl from IMDB and schedules the content.

Step 3: The scheduler will return the next URL's to crawl the website and sends them to downloader middleware.

Step 4: After the desired content is downloaded then it converts them to JSON format which is shown in the code.

Step 5: It extracts the required content of the data that is in the JSON format by using the tags within the header.

Step 6: The obtained data is then stored across the various dimensions of user's choice and prescribed in data frames facilitated by pandas module in python.

Step 7: Now visualizations are drawn across any two dimensions to formulate a business oriented data driven decision.

Step 8: Now we can rapport the relationships across the film's attributes and derive conclusions on the famous genres on a given particular year.

Fig. 2. Web Scraping pipeline Architecture

4. Results

Step1: We will append the score column based on computation over existing attributes is displayed below:

Table 4. Scores embedding for attributes

Sno	Name	Genre	Rating	Score
0	Army of the Dead	Action	5.9	0.000059
1	Army of the Dead	Crime	5.9	0.000059
2	Army of the Dead	Horror	5.9	0.000059
3	Cruella	Comedy	7.4	0.000256
4	Cruella	Crime	7.4	0.000256

Step 2: We have visualized the movies as per rating and we can see how matplotlib plots the graph:

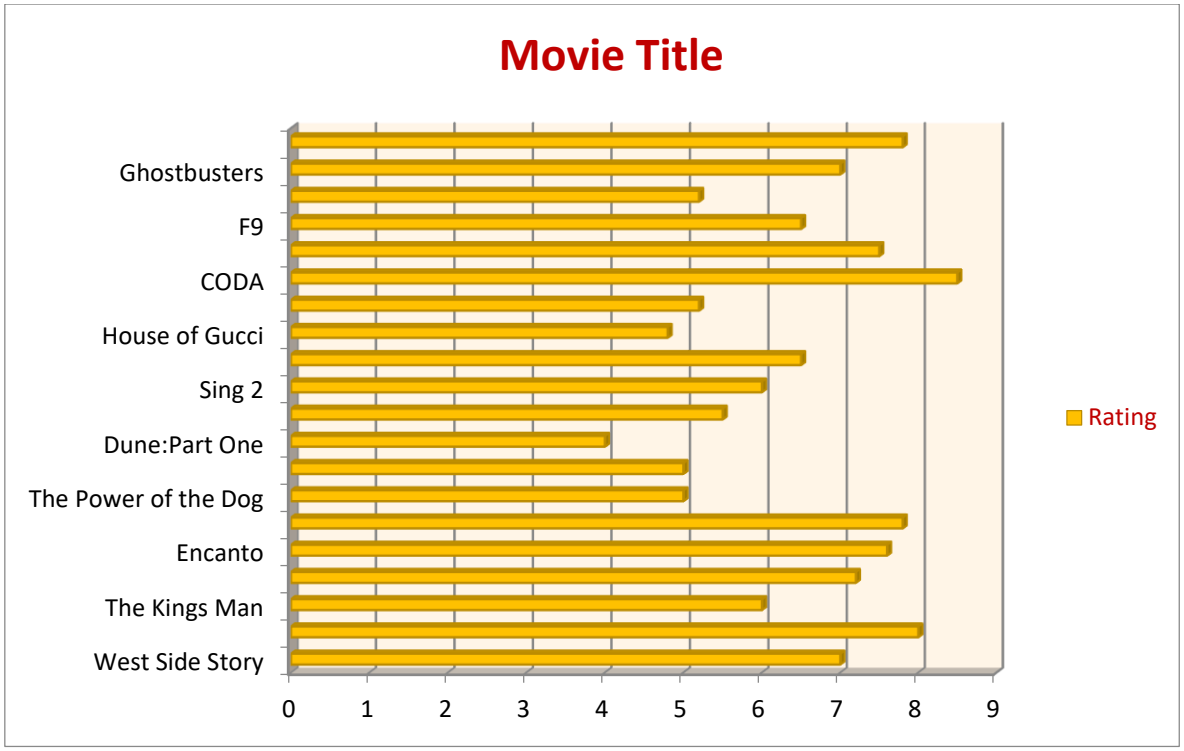

Fig. 3. Visualizing Movie Rating

Step 3: Generating the Runtime, Certificate , Review of each Movie .

We have used Matplotlib, which is an amazing library based on Python for 2D layout episodes. Matplotlib is a multi-displinary platform data visualization library built into the NumPy array and developed to work with a wid range of SciPy stack. It was launched by John Hunter in 2002.

One of the great advantages of visualization is that it allows us to access huge amounts of data on amicably digestible mirrors. Matplotlib contains many sites like scatter, bar, line, histograms etc.

Table 5. Runtime, Certificate, Review of Each Movie

S.No	Id	Movie Title	Genre	Rating	Certificate	Reviews
0	tt0993840	Army of the Dead	[Action,Crime,Horror]	5.9	18	99788
1	tt3228774	Cruella	[Comedy,Crime]	7.4	PG-13	28905

2	tt11083552	Wrath of Man	[Action,Thriller]	7.2	R	35696
3	ttt5433138	F9	[Action,Adventure,Crime]	5.2	PG-13	5620
4	tt6111574	The Woman in the window	[Crime,Drama,Mystery]	5.7	18	42815
5	tt3215824	Cruella	[Action,Drama,Thriller]	6.0	R	25292

Step 4: Generating Visual Results

We have used a histogram, which is basically used to represent data provided by a particular type of groups. It is an accurate representation of the image distribution of numerical data. It is a type of bar structure where the X-axis represents the width of two while the Y-axis provides information about frequency. To create a histogram the first step is to create a width bin, then distribute the entire range of values in the interval series, and calculate the values that fall in each interval extreme frequency variables. The function of matplotlib.pyplot.hist () is used to calculate and create x histogram.

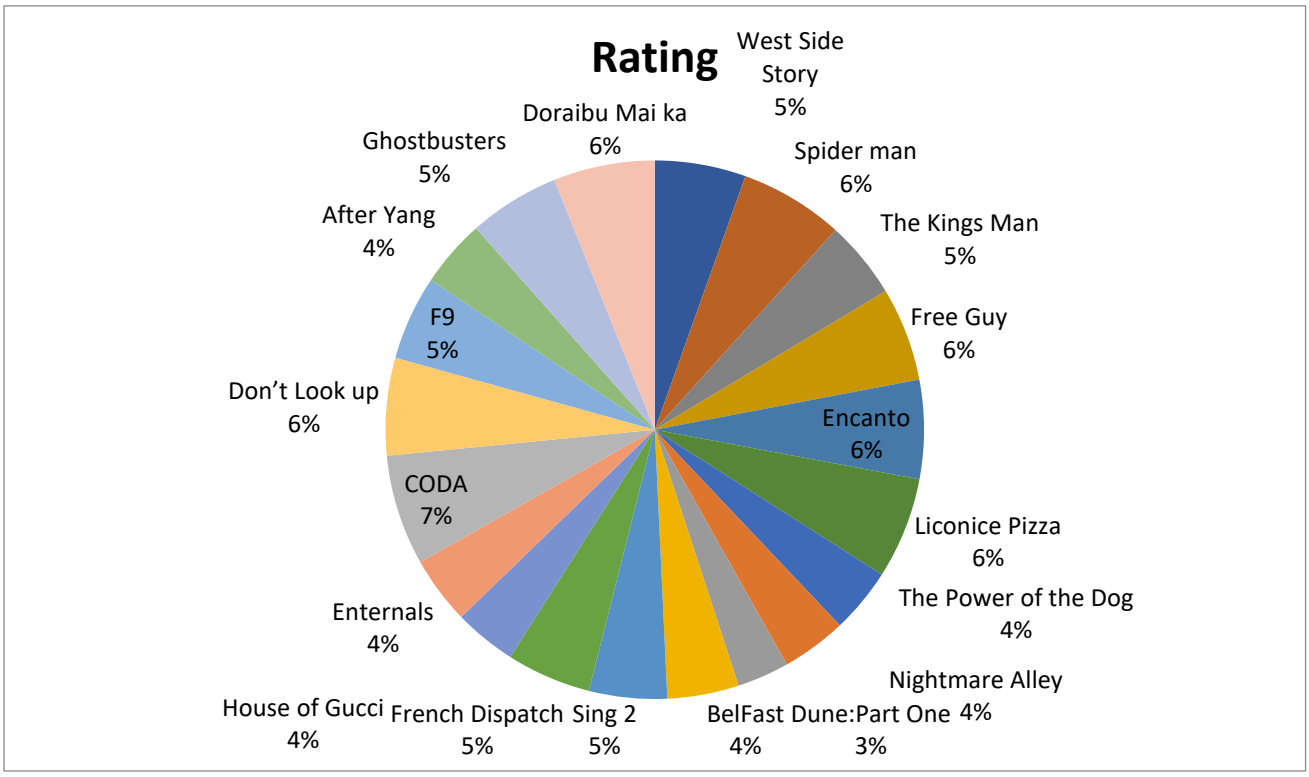

Fig. 4. Different Visual Representation for Movie Rating

Step 5: Handling Complex Data Structures

Handle complex data structures is a vital task. Coding can be done and then we can decrypt JSON manually, but we can get rid of this work with a much smarter solution. Instead of switching from custom data type converted JSON, you can forego on mediator step. All that we have take-up is to represent your data in terms of the built-in types that JSON understands. In fact, we translate complex material into simple presentations, in which the json translates all models into JSON format.

Table 6. Evaluation for data in JSON Format

S. No	Name	ContentType	Genre	Rating	No of Reviews	Score	Director	No of reviews	S
0	Army of the Dead	18	[Action,Crime, Horror]	5.9	NAN	NAN	Zack Snyder	98067.0	0.000060
1	Cruella	PG-13	[Comedy,Crime]	7.4	NAN	NAN	Craig Gillespie	26012.0	0.000284
2	Wrath of Man	R	[Action, Thriller]	7.2	NAN	NAN	Guy Roitchie	34063.0	0.000211

Now that we have seen how we load our JSON data into our Python application, let's take a look at the options we have to work with JSON.

• Extracting information from JSON is as simple as defining a method we follow and providing a keyword pairing name.

• If you are already comfortable with Panda and would like to work with flat data, there is a faster way to get your data into data frames. First, you can take the first level of your data and add it to the data frame.

5. Conclusion and Future Work

Extrusion of hidden web data has been a big issue in recent years due to the independence and diversity of hidden web material. Search engines have now become an inefficient technique of finding this type of data. An easy-to-use search interface, identification, query analysis, and effective methods of extracting data based on web design, form submission analysis, and a new delivery system are the key achievements of this paper. The unequal structure of the web makes it difficult as web is dynamic environment with inefficiencies in organizations and knowledge structure. There is no regulation to follow when building web proximity. Python is used as it consistently gets richer and as the number of data scientists continues to surge. As we get in deep with machine learning, in-depth learning, and other related data science activities, we will probably see these improvements available for our use as Python libraries.

We can also rest our services via an AI driven application such that when any website is entered might be domain name or else the absolute path directing to the specified website to be scraped. With that being said this application that automatically takes the input of web site and also parameters to be tuned for the outcome if given should be able to able to provide business decisions and also the computation needed to be done is to be done in backend processing of the application. Thus, my further scope of extending this project will be in the machine learning domain to further automate the system and expect fruitful and accurate decisions out of it.

Refrences

[1]. E. Uzun, "A Novel Web Scraping Approach Using the Additional Information Obtained From Web Pages," in IEEE Access, vol. 8, pp. 61726-61740, 2020, doi: 10.1109/ACCESS.2020.2984503.

[2]. D. M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 450-454, doi: 10.1109/ICECA.2019.8822022.

[3]. M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” J. Big Data, vol. 2, no. 1, p. 1, Dec. 2015.

[4]. E. Uzun, H. V. Agun, and T. Yerlikaya, “A hybrid approach for extracting informative content from web pages,” Inf. Process. Manag., vol. 49, no. 4,pp. 928–944, 2013.

[5]. E. Uzun, E. S. Güner, Y. Kılıçaslan, T. Yerlikaya, and H. V. Agun, “An effective and efficient web content extractor for optimizing the crawling process,” Softw. - Pract. Exp., vol. 44, no. 10, pp. 1181–1199, Mar. 2014.

[6]. E. Uçar, E. Uzun, and P. Tüfekci, “A novel algorithm for extracting the user reviews from web pages,” J. Inf. Sci., vol. 43, no. 5, pp. 696–712, Sep. 2016.

[7]. Y. C. Wu, “Language independent web news extraction system based on text detection approach,” Inf. Sci. (Ny)., vol. 342, pp. 132–149, May 2016.

[8]. E. Uzun, H. N. Bulu ̧s, A. Doruk, and E. Özhan, “Evaluation of Hap, An-gleSharp and HtmlDocument in web content extraction,” in International Scientific Conference’2017 (UNITECH’17), 2017, vol. 2, pp. 275–278.

[9]. H. N. Bulu ̧s, E. Uzun, and A. Doruk, “Comparison of string-matching algorithms in web documents,” in International Scientific Conference’2017(UNITECH’17), 2017, vol. 2, pp. 279–282.

[10]. F. E. Grubbs, “Procedures for Detecting Outlying Observations in Sam-ples,” Technometrics, vol. 11, no. 1, pp. 1–21, 1969.

[11]. F. E. Grubbs, “Procedures for Detecting Outlying Observations in Samples,” Technometrics, vol. 11, no. 1, pp. 1–21, 1969.

[12]. E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, “Web data extraction, applications and techniques: A survey,” Knowledge-Based Syst., vol.70, pp. 301–323, Nov. 2014.