2024 International Conference on Communication, Computing and Internet of Things (IC3IoT) | 979-8-3503-5268-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/IC3IoT60841.2024.10550225

A Comprehensive WebScraping of IMDb's Top 50 Movies using Beautiful Soup

1st Akhilan Anbu
B.Tech. in Computer Science and Engineering (CORE)
School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT), Chennai, and
Intern, Computational Intelligence Research Foundation(CIRF),
Chennai
Email: akhilananbu@gmail.com

2nd Doreen Hephzibah Miriam.D
Director,
Computational Intelligence Research Foundation(CIRF),
Chennai, India.
Email: director@cirf.co.in

3rd C.R.Rene Robin
Department of CSE,
Sri Sairam Engineering College, Chennai, India.
Email: crrenerobin@gmail.com

***Abstract*—Data encompasses all factual and measurable information that is accessible, quantifiable, or recordable. This information can take on a variety of forms, including numerical figures, written content, visual depictions, or symbolic annotations. The primary goal behind this undertaking is to extract such data from the official website of the Internet Movie Database, IMDb's top 50 movies of all time using the BeautifulSoup package available in the Python package library which can be used for HTML parsing to create a parse tree for parsed pages. These parsed pages can in turn be utilised to extract data from the HTML code snippets used to code the official website's data aiding in further analysis and research studies. For the main objective here, we have used IMDb's official website as the foundation, with the language Python serving as a programming link to scrape data using the built-in functions in the Python package library such as Numpy, Pandas, Requests and BeautifulSoup.**

In this paper, we create an accurate Data Frame using a Python package called Pandas, this Data Frame can enable users to search for any required attribute from the variety of attributes available in the data frame such as release year, meta scores of each film, the type of genre they belong to and parental guidance suggestion. The objective of the paper mainly focuses on easing the user's search process by providing them with different attributes they can select from. The example website utilisation presented herein serves as a fundamental illustration of our research efforts. Extrapolating this methodology across various websites spanning multiple domains opens up an immense amount of opportunities for insightful analysis and strategic data utilization, considering the transformative potential of employing such a thorough strategy across a range of digital platforms.

***Keywords:* - BeautifulSoup, Pandas, Numpy, HTML, IMDb, Requests**

I. Introduction

Over the last few years, Technology has developed drastically in such a way that we are able to attain any amount of data and any type of data in a short while. Datasets are available everywhere and by making use of the internet we happen to have it at the grasp of our hands. The primary issue at the moment is that we do not have enough knowledge on how to manipulate such data with ease. Each data has its own attribute types, keywords, Data Types, Subject area, tasks, feature types, Instances, etc. And our talent is to find the right amount of data with the right attributes. The quality, Correctness and quantity of the data play a major role in the manipulation of the required data to attain the right products. So, we have taken IMDb's official website for web scrapping Since IMDb is the world's most authoritative and popular source for movies, TV shows and celebrity content, particularly made and designed for fans to traverse the world of Features and Cinema.

In the present age of digital technology, there has been an extraordinary increase in the volume of data being generated. Data science is dedicated to the collection, analysis, and interpretation of data in order to extract insights and enable well-informed decision-making. [15] Communication technology has improved globally, particularly since the creation of the World Wide Web. This creates an appropriate setting for online transactions like banking via the web, payment through the internet, and shopping online. [17]

Web scraping selects data automatically and displays it in a format we can make sense of easily. Web scraping can be used in a wide variety of situations but here we use it for the collection of specific movie data. If you're an avid movie enthusiast, getting movies to watch every day can be a pain, especially when the information you need is found across several web pages. We'll make data extraction easier by building a web scraper to retrieve data on movies automatically from the Internet. Rather than meticulously copying and pasting information manually, we can develop scripts to automatically retrieve and organize data.

979-8-3503-5268-9/24/$31.00 ©2024 IEEE

A. Literature review

In general, data from web scraping is the methodical technique for obtaining and aggregating interesting content found on the Web. A computer intermediary, commonly referred to as a Web robot, simulates the surfing exchange among individuals and website servers in a traditional Web traversing process. [4] We found that the majority of web scrapers are generally very similar and made to carry out usual, easy tasks. Stated differently, they might not seem to be as adaptable and all-encompassing as you might anticipate. Although every developer of web scrapers tries to have their program retrieve any kind of website, we have found that a few web scraping applications are more appropriate for certain tasks than others. Classic copy-and-paste method: Copy-and-paste with traditional evaluation is typically perhaps the most efficient and useful internet scraping method.[16] Nevertheless, it is an error-prone, time-consuming, and uncomfortable process whenever clients have to discard a lot of datasets.

Web scraping has generated controversy, though, since individuals have copied entire sites and declared the content as theirs to use, which has led to legal action and copyright violations. Any business or organization engaging in web scraping must be closely watched or put through a plagiarism detection procedure in order to prevent any more disputes down the road. [6]. We ought to acquire an awareness of the size and arrangement of the desired web page before we start scraping it. [9] Nonetheless, more needs to be done to bolster privacy safeguards and inform people and companies regarding the dangers of data violations.[18]

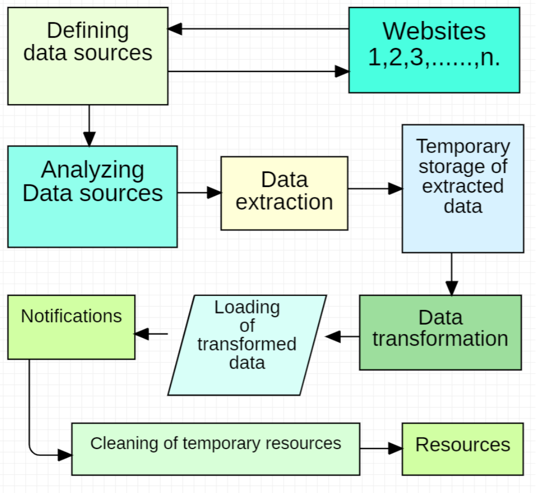


Fig. 1: data extraction transition in webscrapping

As shown in Fig.1, Examining the effectiveness of alternative web scraping techniques, including the HTML selector, vertically grouping techniques, semantic analysis and recognition, and automated web page research is one possible avenue for future research. To optimize the prior approach, more variables can be added when testing, fixing, or integrating techniques to address the shortcomings of current methods. [5] IMDb frequently changes the design and organization of its website, which makes scraping difficult. To account for these changes, researchers frequently need to modify their scraping scripts. Due to discrepancies in the data retrieved, such as missing data or inaccurate data, data cleaning and preprocessing are also essential steps.

By incorporating cutting-edge machine learning and artificial intelligence algorithms, tools for scraping websites can grow more intelligent and self-sufficient. They would be better able to comprehend website frameworks, adjust to modifications, and gather pertinent data even from intricate, ever-changing web pages. Web scraping resources may develop for gathering information from videos, photos, and other types of forms of media as the internet grows increasingly visually focused.

II. Applied tools

Relevant tools used to web scrape data:

A. Python

Python is a powerful, elegant language used for programming that is easy to read and understand.[13] It shows almost all of these characteristics are similar to many of the other languages and are valid for corporal applications. Moreover, Python is versatile in its support for diverse programming paradigms, spanning functional and object-oriented programming (OOP) methodologies. Python's design places a deliberate emphasis on the principles of code clarity, comprehensibility, and sustained manageability. In the end, Python was opted for this project as the preferred programming language, primarily due to its rich repository of specialized libraries tailored for data processing, analysis, and visualization purposes. Even though Python came into existence only recently around 30 years ago in 1991, its use was found incredible among data analysts and data scientists because of the ease in data extraction when they can be imported effortlessly with Python packages downloaded from their respective libraries.

B. Beautiful Soup

Beautiful Soup, an essential library in Python, is utilized for web scraping and the extraction of data from HTML snippets and XML docs. It streamlines locating tags, attributes, and content within these documents. Its proficiency in managing unorganized and poorly structured XML and HTML has led to its recognition as a valuable asset in web data extraction. The library provides an array of valuable classes, comprising Beautiful Soup, which can streamline the parsing of HTML docs. This study will demonstrate the library's efficacy & versatility, which are highly regarded by data analysts and developers for their web scraping undertakings. Hence, It is much better to practice specifying the parser that will be used while making a BeautifulSoup object. This is due to the fact that different parsers parse the content differently.[11] This is most obvious in cases where we input an HTML statement which is not valid to parse.

C. Pandas

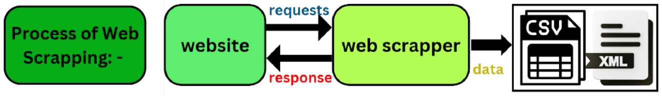
Pandas is a Python package that aims to work with extensive available data sets. It can be used for manipulating, extracting, analysing, cleaning and exploring data sets using the required functions available in the Python library. Pandas, an open-source library designed for Python, plays a pivotal role in simplifying data manipulation and analysis. It furnishes a set of data structures optimized for the efficient storage and manipulation of structured data, akin to tables with rows and columns. Inside the Pandas framework, you'll encounter two fundamental data structures: Series and Data Frame.
Furthermore, it equips users with a toolkit for in-depth data analysis, handling time series data, and seamless Writing and reading information in several file types, such as CSV, Excel, and SQL databases. It encourages the widespread use of Pandas in computational science, data analysis, and other disciplines that call for the manipulation and evaluation of data. [15]

D. Power Bi

This instalment of the Strongest Tools in the Build section presents Power BI, an application from Microsoft, along with its corresponding features integrated into Excel versions 2013 and later. IT librarians, among others, can utilize Power BI to aggregate, examine, illustrate, and distribute information gathered from an extensive range of sources of data used in library administration. Excel is used in almost every kind of office setting, and the advantages of assimilation are extensively covered in the libraries. [2] The graphs presented here were created using Microsoft's Power BI, an extremely efficient tool that helped us comprehend and analyse data greatly. It can Develop datasets from any source of data and easily add that data into the Power BI information centre to establish a centralized, single point of information for all your data. In our usual Google Colab environment, we can use sensitivity designations to Power BI information using Microsoft Data Security. Thanks to Microsoft Cloud App Security, you can provide lots of data avoidance and authority to Power BI users, involving report distribution.

III. Web Scrapping

Detailed description and algorithms used for Web Scrapping:



A. Web Scrapping

Web scraping is a method of automatically obtaining a lot of data via websites. The majority of this information is unorganized HTML data that is transformed into organized information in a database or spreadsheet before being utilized in different applications. Specific data or all the data required can be obtained through web Scrappers. For collecting info from websites, web scraping can be done in a variety of ways. Web scrapers may collect all of the data on a specific site or any information that an individual desires. You should specify what information you require to ensure that the web scraper gathers only that data. Web Scrapping provides you with a method to obtain selected data from the website using Python easily. Before scrapping the required data from a website, the site's URL should be predisposed. Doing this will enable it to capture specific data from the required website from where it extracts the data.

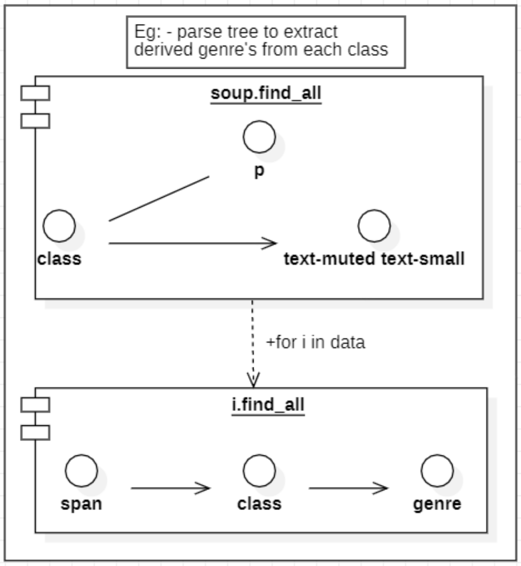


Fig. 2: Extract genres list

With top-of-the-statistical tools powered by AI and an accessible clicking-and-dropping report canvas, you can transform unprocessed information into engaging visuals. It transforms knowledge into choices faster by incorporating perspectives into applications you already use, such as Google Colab, while also protecting your data. Pandas extends its support to a broad spectrum of data processing handling, encompassing operations like data selection, extraction and transformation, as well as the integration and grouping of Data Sets. The website's structure must then be brainstormed and analogised and the selective data for extraction should be identified. The required attributes and the tags that are given in the HTML or XML code of the website are to be singled out from the rest of the code snippets. After associating and connecting the data, the program code will connect to the website's URL, as shown in Fig.2, extracts the required data and save the data in the format provided by the user.

B. Operating a Web Scrapper

Before making a Web Scrapper, ensure that the last updated version of Python has been installed. Executing this command statement will set up the most recent and updated version of BeautifulSoup and its packages. The installation can be verified using the command ”from bs4 import beautifulSoup”. The procedure of extricating data using automated tools by making use of scripts from the websites that can automate the GET request to the required website’s URL, extract its data and store the data in an organised format specified by the user is called Webscapping. Finding the best possible combination of feature weights is crucial when implementing categorization [19]There are several important steps to running a web scraper. First, learn the fundamentals of HTML as well as CSS because web scraping relies on parsing these file types. Learn how to code in a language such as Python and install necessary libraries like requests and BeautifulSoup.

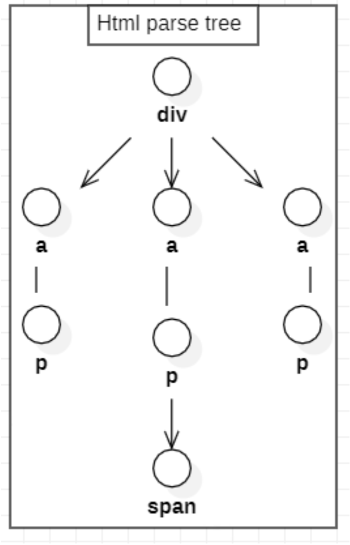


Fig. 3: HTML code parse diagram

Use the developer tools in your browser to examine the target website and find the code tags in HTML classes that hold the desired information. Select the right library based on the complexity of your project; Scrapy provides a more robust framework, while BeautifulSoup is better suited for simpler tasks. Create your scraper by requesting HTML content from the website via an HTTP request, analyzing it to gather pertinent information, and then writing your script. If necessary, handle dynamic content with tools such as Selenium and develop pagination logic if the data is spread across multiple pages.

C. Steps to Web Scrap Data

1) Initiate access to Google Colab or an analogous application designed to facilitate web scraping procedures.
2) Establish a novel notebook, designate an apposite appellation to your Colab notebook, and introduce a fresh code cell to commence the process.
3) The preliminary stride in constructing a web scraper involves the systematic importation of essential Python package libraries.
4) Integrate pandas, numpy, and requests packages from the Python library, employing the directive ’from bs4 import BeautifulSoup’ to seamlessly import the BeautifulSoup package.
5) Employ the requests package to execute the ”get” function, supplying the URL of the targeted website.
6) Apply the BeautifulSoup package and execute the ”prettify” function to procure the HTML-formatted representation of the website’s code as shown in Fig.3, ensuring optimal indentation.
7) The subsequent imperative in effective web scraping necessitates discerning the classes and objects embedded within the HTML code of the website, specifically those germane to the desired data extraction.
8) Retrieve the classes and objects within the website’s code by means of a judicious right-click and subsequent selection of the ”inspect” option. Alternatively, leverage the Inspector Gadget extension, available in Google extensions, to achieve the same objective.
9) Instantiate an empty list, poised to serve as a receptacle for the storage and manipulation of the harvested data.
10) In instances where the website’s URL spans multiple pages, employ the range function to systematically traverse each page, facilitating the comprehensive collection of requisite data.
11) Subsequent to this traversal, invoke the FindAll function from the BeautifulSoup package to systematically acquire the targeted data.
12) Append the retrieved data to the previously initialized empty list, affecting the aggregation of all selected data within the designated list.
13) In scenarios where data types within the website vary, employ the ”extend” function in conjunction with the ”text” function to distil and extract textual content from the code, thereby ensuring a uniform data type within the list.
14) With data successfully extracted, scrutinized and cleansed into a coherent textual format, the ensuing step entails methodically organizing the data into a meticulously structured data frame.
15) To facilitate this organization into a data frame, leverage the Pandas package from the Python library, harnessing the versatility of the DataFrame function for seamless accessibility.
16) Import the Pandas library and instantiate separate data frames for each requisite attribute, employing the DataFrame function judiciously.
17) Utilize the Concat function to amalgamate these distinct attributes into a comprehensive Data Frame, wherein extracted and cleansed data coalesce seamlessly. Specify ’axis=1’ to concatenate Dataframe objects horizontally along the x-axis.
18) Subsequently, all collated data is amalgamated into a refined data frame, poised for thorough analysis, exploration, and manipulation, having been sourced directly from the website’s URL through adept web scraping techniques.
19) Leverage the filter option to selectively distil attributes

such as genres, meta scores, release years, gross amounts, and parental guidance recommendations according to your specific requirements.

20) Revel in the cinematic insights gleaned from the meticulously crafted data frame, extracted adeptly from the website, fostering a heightened proficiency in data-driven exploration.

Pseudocode assists you in clarifying the URL of the site you've been scraping and the particular information you're looking for. This sets the foundation for your scraping algorithm. This could include using a library such as BeautifulSoup or a different HTML parsing tool as shown in Fig.4, It helps plan how you will use the HTML structure to find the parts containing the data you need. Pseudocode helps through the process of creating a loop for iterating through the identified parts and extracting relevant data from each You can use pseudocode to simplify the complex process of web scraping by breaking it down into smaller steps, thereby making it easier to develop, plan, and execute your scraping algorithm in a particular programming language.

D. Detailed Pseudocode: -

```
#import pandas, numpy, and requests
#from bs4 import BeautifulSoup
Algorithm_Web Scrapping()
    outcome <— requests.get(”URL”)
    soups <—BeautifulSoup(outcome.content, ’HTML.parser’)
    print(soup.prettify())

#sample algorithm for metascores: -
Algorithm_append_metascores()
    metascore <—[ ]
    aa <–[ ]
    data <– soup.find_all()
    for i in data:
        ab<–i.find_all()
        aa.extend(ab)
        for h1 in aa:
            metascore.append(h1.text)

#similarly for each of the other attributes, create empty lists and append data to the list
#To make a detailed data frame: -
#import pandas package

Algorithm_get_dataframe()
    df1 <— pd.DataFrame(’metascore’:metascore)
    #similarly df2, df3.... for other attributes
    data_fr <—pd.concat([Df1, Df2,........,Dfn], axis=1)
end
```

The preceding algorithm's operation is discussed in the subsection that follows, which deals with the evaluation of performance, disadvantages or challenges encountered whilst developing the web scrapper, the intended website's initial outlook, and the final output observed.

IV. Performance Evaluation

Discussing the performance of our structured program:

A. Data Analysis

Using the Describe function from Pandas Library, we can take a look at the information statistics such as count, unique, top and frequency for each generated attribute, particularly parental guidance, meta scores, genre, gross rate and release year. We know that there are more statistics, data and facts on the Internet than that of which a Human being can take in and learn per lifetime. What you need is not access to that information, but a scalable way to collect, organize, and analyze it.[20] Gaining knowledge of the technique is necessary in order to arrive at the various methods that are used in this work.[1] The fundamentals of which include manipulating several parameters that are utilized to make an endpoint request appear valid. When a basic HTTP request is sent to a website, the majority of them often produce a valid answer; However, a lot of those have systems in place to distinguish a legitimate request from an inquiry conducted by an AI bot. This study examines many online scraping strategies, ranking them from poorest to highly effective in terms of information collection.

Numerous situations can benefit from online scraping, including customer scraping, evaluations of product collecting, price fluctuation tracking and contrast, property list storage, climate evaluation, site modification detection, and internet information incorporation. [21] The way in which we scrape the web page will depend on how big it is.[9] Efficiency is meaningless for a site with a couple of hundred URLs. It would require weeks to save every page of a website

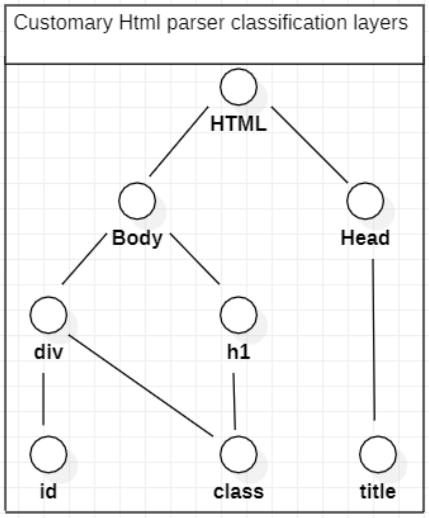


Fig. 4: Fig representing HTML parse layers

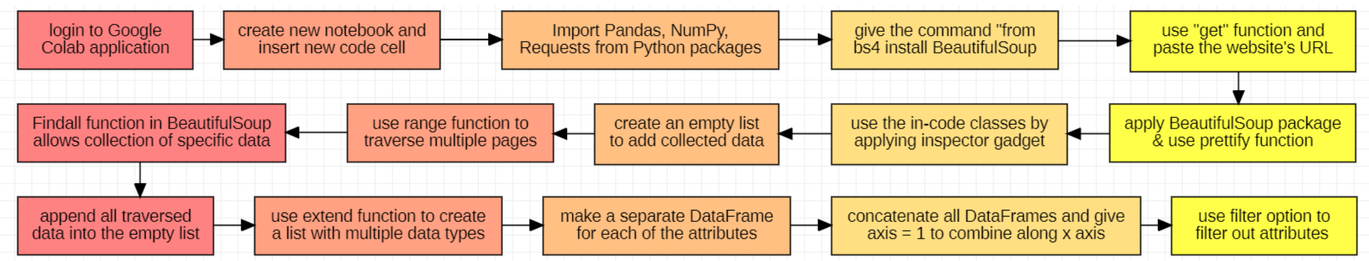

login to Google Colab application
create new notebook and insert new code cell
Import Pandas, NumPy, Requests from Python packages
give the command "from bs4 install BeautifulSoup
use "get" function and paste the website's URL
apply BeautifulSoup package & use prettify function
use the in-code classes by applying inspector gadget
create an empty list to add collected data
use range function to traverse multiple pages
Findall function in BeautifulSoup allows collection of specific data
append all traversed data into the empty list
use extend function to create a list with multiple data types
make a separate DataFrame for each of the attributes
concatenate all DataFrames and give axis = 1 to combine along x axis
use filter option to filter out attributes

Fig. 5: A Step by Step process to constructing a Web Scrapper.

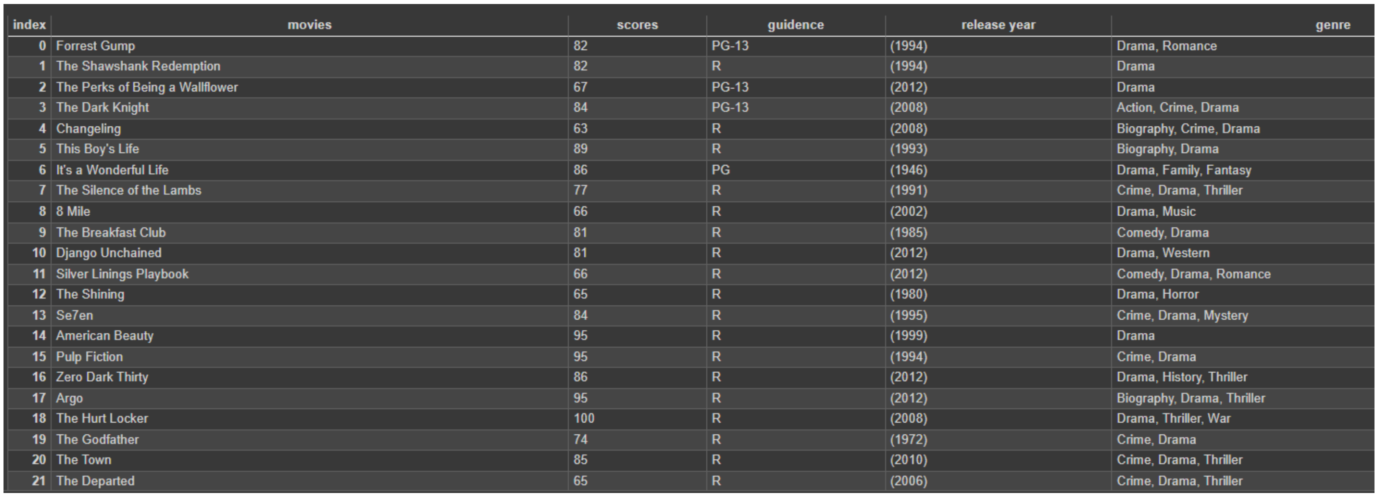
with more than one million sections, though. Examining the findings of Google's crawler that of which had probably recently scanned the page we are keen on is a fast approach to gauge the size of a website. Statistics may then be used to display the data that was gathered via web scraping approaches. Python programming is used to build data metrics. Data categorizing helps with the processing of information as required. [7]

IMDb regularly modifies the layout and appearance of its official website, making scraping challenging. Scholars often need to adjust their scraping scripts to consider these changes. Cleaning the information and preparing are also necessary procedures because of inconsistencies in the obtained data, such as removing incorrect data. One potential direction for future research is to examine the efficacy of different possible web scraping methods, such as automated web page research, keyword analysis and recognition, vertically grouping techniques, and the HTML selector. Additional factors can be introduced when evaluating, integrating, or repairing methods that tackle the drawbacks of existing methods in order to maximize the prior approach.

B. The Target Website:-

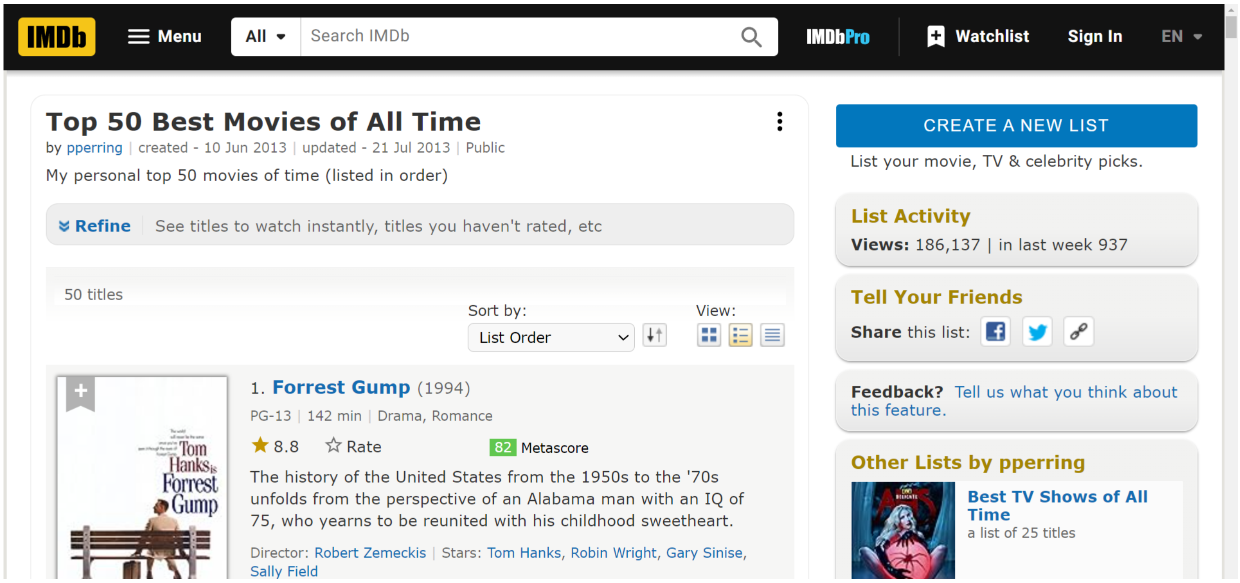
The system in place allows users to select the website of their choice, identify the data to be scraped, create a code to extract data using the website's URL, run the session using an HTML parser, and then analyze the scraped information from a neatly structured data frame. Here, we can see how the web scraper has converted the website as depicted in Figure 7 and The website's HTML code into a well-organized data frame, making it simpler for us to obtain information directly.

DataFrames match data effortlessly depending on its index and the column tags. This implies that DataFrame tasks typically occur component-wise, so you don't have to be concerned about manually arranging the information. DataFrames are capable of handling different data types, such as mathematical, categorized and written data. This adaptability is essential when dealing with actual data sets, which frequently include a variety of data types. Figure 6 depicts the results of our work. DataFrames provide a flexible yet effective way to interact with structured information by offering a variety of resources and techniques for analyzing data and data manipulation. The contents and labels from the pandas DataFrame can be saved and loaded to form a variety of file formats, which include CSV, Excel, SQL, and others. These files are subsequently used to aid us with the visualisation part of the data analysis. In our case, we loaded the extracted CSV data file into Microsoft's Power BI application which can generate multiple visualisation graphs, which is further discussed in the subsection below.



index	movies	scores	guidence	release year	genre
0	Forrest Gump	82	PG-13	(1994)	Drama, Romance
1	The Shawshank Redemption	82	R	(1994)	Drama
2	The Perks of Being a Wallflower	67	PG-13	(2012)	Drama
3	The Dark Knight	84	PG-13	(2008)	Action, Crime, Drama
4	Changeling	63	R	(2008)	Biography, Crime, Drama
5	This Boy's Life	89	R	(1993)	Biography, Drama
6	It's a Wonderful Life	86	PG	(1946)	Drama, Family, Fantasy
7	The Silence of the Lambs	77	R	(1991)	Crime, Drama, Thriller
8	8 Mile	66	R	(2002)	Drama, Music
9	The Breakfast Club	81	R	(1985)	Comedy, Drama
10	Django Unchained	81	R	(2012)	Drama, Western
11	Silver Linings Playbook	66	R	(2012)	Comedy, Drama, Romance
12	The Shining	65	R	(1980)	Drama, Horror
13	Se7en	84	R	(1995)	Crime, Drama, Mystery
14	American Beauty	95	R	(1999)	Drama
15	Pulp Fiction	95	R	(1994)	Crime, Drama
16	Zero Dark Thirty	86	R	(2012)	Drama, History, Thriller
17	Argo	95	R	(2012)	Biography, Drama, Thriller
18	The Hurt Locker	100	R	(2008)	Drama, Thriller, War
19	The Godfather	74	R	(1972)	Crime, Drama
20	The Town	85	R	(2010)	Crime, Drama, Thriller
21	The Departed	65	R	(2006)	Crime, Drama, Thriller

Fig. 6: Derived and cleaned final Data Frame



IMDb Menu All Search IMDb IMDbPro Watchlist Sign In EN

Top 50 Best Movies of All Time

by pperring | created - 10 Jun 2013 | updated - 21 Jul 2013 | Public

My personal top 50 movies of time (listed in order)

Refine | See titles to watch instantly, titles you haven't rated, etc

50 titles

Sort by: List Order

View:

1. Forrest Gump (1994)

PG-13 | 142 min | Drama, Romance

★ 8.8 ☆ Rate 82 Metascore

The history of the United States from the 1950s to the '70s unfolds from the perspective of an Alabama man with an IQ of 75, who yearns to be reunited with his childhood sweetheart.

Director: Robert Zemeckis | Stars: Tom Hanks, Robin Wright, Gary Sinise, Sally Field

CREATE A NEW LIST

List your movie, TV & celebrity picks.

List Activity

Views: 186,137 | in last week 937

Tell Your Friends

Share this list:

Feedback? Tell us what you think about this feature.

Other Lists by pperring

Best TV Shows of All Time

a list of 25 titles

Fig. 7: Target Website: IMDb's Top 50 Movies of All Time (from which data was extracted)

C. Visualisation of the Data Frame

Data analytics includes information from statistics analysis. NumPy is the foundational Python package for computational science. Numpy is used during this study to perform operations such as average, standard deviation, count, highest, lowest, tail, and head. In this stage, statistical analysis of data was performed on a CSV file containing movie information from a website obtained through our constructed web scraper. A thorough knowledge of Numpy will allow you to maximise resources like Pandas.

A pie chart in Fig.9 is a visually appealing tool for illustrating the variation of meta scores between a group of movies. Meta scores, which are frequently compiled from multiple critiques with multiple critical evaluations, provide a reinforced analysis of a film's overall acceptance. This pie chart was made with the help of Microsoft's Power BI application which depicts how those ratings are distributed among various movies in a clear manner, offering audiences a brief but beneficial image of the overall evaluation of the films under assessment.

The pie chart is separated into cuts, each of which corresponds to a different movie. Each slice's size corresponds to the meta score given to the corresponding movie. Movies that have greater meta scores have bigger slices, suggesting better critical acceptance. Films with lower ratings will have smaller slices, indicating a less positive assessment which is depicted in Figure 9. To improve the visual experience, colour coding can be used, with warmer colours indicating higher scores and cooler colours representing lower scores. This colour gradient aids viewers in quickly identifying the range of positive reviews within the dataset.

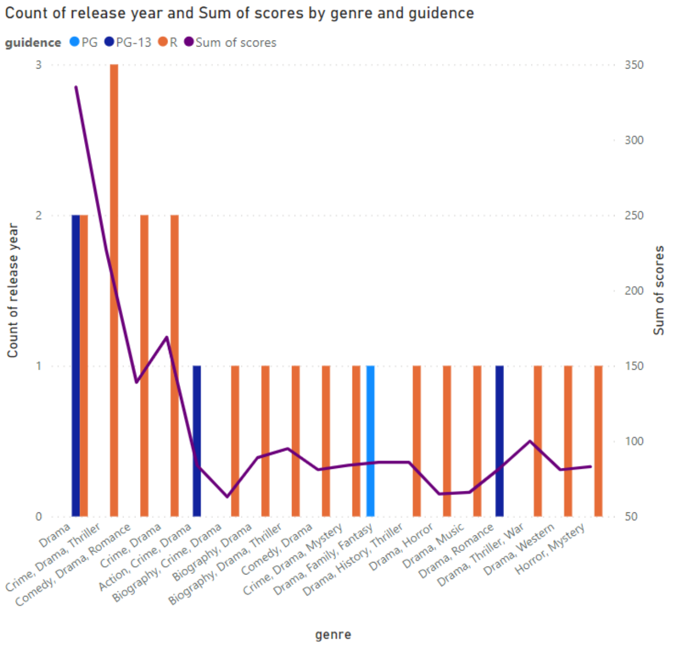


Fig. 8: Visualising a line and clustered column chart

The figure 8 depicts a line and clustered column chart with genre types in its X-axis, the release years in its column y-axis, a sum of scores in its line y-axis, and a column legend with parental guidance suggestions. Using the Power BI application has provided us with tools for visualizing the organized data frame, from which we conclude that the most recent movies screened from 2000 to 2023 with Parental guidance R rating and genres ranging from crime, and drama to thrillers have received an expansive amount of rating compared to the movies released prior to 2000.

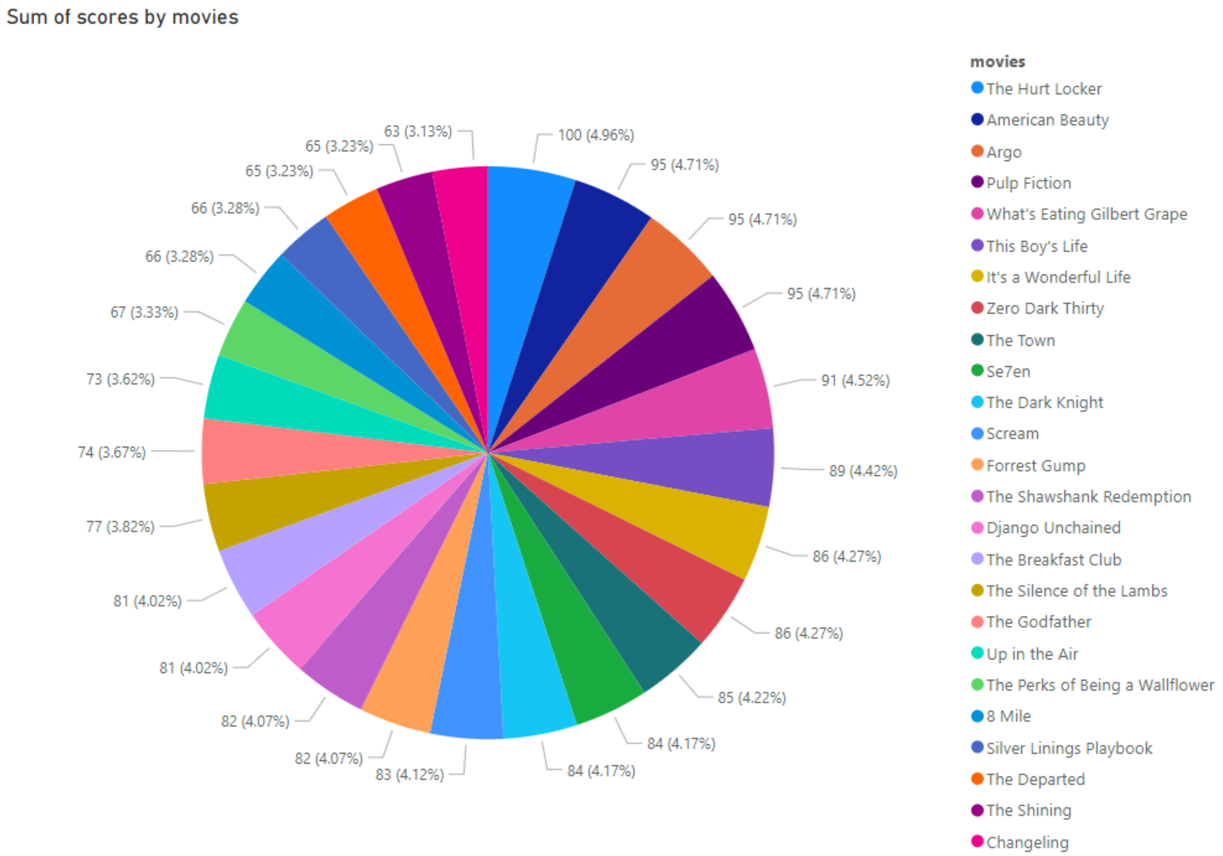


Fig. 9: Pie chart aiding Visual Representation for Movie Meta-Scores

D. Challenges to Web Scraping

Although Web Scrapping enables ease of data collection and extraction there are certain disadvantages or challenges to it. There are five main challenges to scrapping data using a web scraper.

The first challenge includes the difficulty of reproducibility mainly because of two reasons. To begin with, the information is dynamic, that is, regardless of whether we scrape the data using the exact same programming, we will receive different outputs and results when the main HTML website's code we are scrapping data from has been updated. (In IMDb's case all the movie's names were changed to a different language and after executing the code again the following day, the movie's names changed back to English).

The second challenge occurs due to not having enough consistency in the websites' structures, these structures alter over the course of time, and the code which is used to extract the selective data might not work anymore. In such cases, we would have to repeat the entire process and we would have to scrape the data again using a different set of classes and objects. The websites are developed and operated independently, resulting in differences in data structure and data organization. This inconsistency may contribute to data being lost or rendering it difficult to extract data consistently.

The next challenge occurs due to a lack of enough data or missing data in the data sets. Anybody who handles data has encountered difficulties dealing with inaccurate and missing information, and numerous algorithms and techniques are being created to handle data when missing information is present. 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 for finding this type of data.[14]

The fourth challenge to Web Scraping is that we have no control over the web. It is crucial to be aware of any possible connectivity and content-related problems. Sometimes when we are working on scrapping data, We wouldn't be able to finish the work if the server that hosts the website went down. It is therefore advised to have backup plans or extensions available for such circumstances. Preserving the website's source code for use in the event of a server breakdown or website error is one such possible fix. Since website servers can only handle a certain amount of demands per minute, they'll either reject inquiries or decrease the rate at which the information can be retrieved.[3]

The fifth challenge is that the web's content is not within our control, and these contents can be divided into two categories which are already coded into the website where the databases are maintained by the website manager versus the data that is being input through the users. It is impossible for a data user to determine whether the exchange originated with the data proprietor or with the individual using the data.[10] Considering IMDb, where the information is related to movies, IMDb's website managers spend more time reviewing users' judgments. But this can in fact be considered as an opportunity to clean data.

V. Conclusion

Given the results, we have successfully replicated all the necessary selective data on the "Top 50 Best Movies of All Time" from IMDb's official website into an organised Data Frame using the BeautifulSoup package and Python as the intermediate language. We have also learned that it is important to look through the website's raw HTML code in order to learn how it was designed and arranged, what type of user data is being gathered, and how that data is accessed. The data may contain different data types that will either be structured or semi-structured. Structured data are data that is thus far categorised into proper components from the source of the material and they can be extracted as it is. While semi-structured data are data that haven't been officially organized by the source of the information provider and have to be additionally processed and organised to classify the data we require. As an example, the release years of the movies were in a structured format and they were readily extracted by using the right classes, while the About section of the movies was semi-structured and required further processing to combine and extract the data into the required form of data type. Analyzing knowledge, humans implemented logical algorithms, networking. Anonymity Network Blocks Unwanted phishing, While websites receive updates, internet scraping tools may malfunction or produce incorrect outcomes, necessitating more monitoring work. The intended website's structure is crucial to web scraping. Updates may be necessary if the structure of the website changes drastically and breaks scraping materials. We have also found out that certain web scraping tools may not be able to manage various character sets and languages, so their assistance for international or multilingual sites may be limited. Lastly, we examine how well different systems for recommendations perform. Next, we evaluate the scheduling computational methods on various, varying-sized data sets. [12]

A system for coding should be created after the conceptual framework of the information source has been created, beginning with the perception of the framework of the final collection of data.[8]. The visualization part of the data holds much importance, before commencing the task. In order to perform web scraping effectively, the users should form specific questions for research related to the data source that would influence the data collected. The scraper's firmware needs to be configured subsequently such that all of these Topics for research inquiries may be put into analysis and explored once the data has been collected.

References

[1] Ajay Sudhir Bale, Naveen Ghorpade, S Rohith, S Kamalesh, R Rohith, and BS Rohan. Web scraping approaches and their performance on modern websites. In *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 956–959. IEEE, 2022.

[2] Louis T Becker and Elyssa M Gould. Microsoft power bi: Extending excel to manipulate, analyze, and visualize diverse data. *Serials Review*, 45(3):184–188, 2019.

[3] Mine Dogucu and Mine Çetinkaya-Rundel. Web scraping in the statistics and data science curriculum: Challenges and opportunities. *Journal of Statistics and Data Science Education*, 29(sup1):S112–S122, 2021.

[4] Daniel Glez-Peña, Anália Lourenço, Hugo López-Fernández, Miguel Reboiro-Jato, and Florentino Fdez-Riverola. Web scraping technologies in an api world. *Briefings in bioinformatics*, 15(5):788–797, 2014.

[5] Rohmat Gunawan, Alam Rahmatulloh, Irfan Darmawan, and Firman Firdaus. Comparison of web scraping techniques: regular expression, html dom and xpath. In *2018 International Conference on Industrial Enterprise and System Engineering (ICoIESE 2018)*, pages 283–287. Atlantis Press, 2019.

[6] Moaiad Ahmad Khder. Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, 13(3), 2021.

[7] Yesi Novaria Kunang, Susan Dian Purnamasari, et al. Web scraping techniques to collect weather data in south sumatera. In *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 385–390. IEEE, 2018.

[8] Richard N Landers, Robert C Brusso, Katelyn J Cavanaugh, and Andrew B Collmus. A primer on theory-driven web scraping: Automatic extraction of big data from the internet for use in psychological research. *Psychological methods*, 21(4):475, 2016.

[9] Richard Lawson. *Web scraping with Python*. Packt Publishing Ltd, 2015.

[10] Hephzibah Miriam, D Doreen, Deepak Dahiya, CR Rene Robin, et al. Secured cyber security algorithm for healthcare system using blockchain technology. *Intelligent Automation & Soft Computing*, 35(2), 2023.

[11] Vineeth G Nair. *Getting started with beautiful soup*. Packt Publishing Ltd, 2014.

[12] Rachel Nallathamby, CR Rene Robin, and Doreen Hephzibah Miriam. Optimizing appointment scheduling for out patients and income analysis for hospitals using big data predictive analytics. *Journal of Ambient Intelligence and Humanized Computing*, 12:5783–5795, 2021.

[13] Why Python. Python. *Python Releases for Windows*, 24, 2021.

[14] Narendra Kumar Rao, Beebi Naseeba, Nagendra Panini Challa, and S Chakrvarthi. Web scraping (imdb) using python. *Telematique*, pages 235–247, 2022.

[15] Le Sang. Web scraping of university rankings and data analysis using python. 2023.

[16] De S Sirisuriya et al. A comparative study on web scraping. 2015.

[17] V SMRITHY SINGH, CR RENE ROBIN, et al. A censorious interpretation of cyber theft and its footprints. *I-Manager's Journal on Information Technology*, 12(1), 2023.

[18] Smrithi Sukesh, CR RENE ROBIN, et al. An analysis of the increasing cases of data breaches in india. *Journal on Software Engineering*, 17(3), 2023.

[19] N Vanitha, CR Robin, and D Miriam. An ontology based cyclone tracks classification using swrl reasoning and svm. *Computer Systems Science & Engineering*, 44(3), 2023.

[20] Justin Yek. How to scrape websites with python and beautifulsoup. *How to Scrape Websites with Python and BeautifulSoup*, 2017.

[21] Bo Zhao. Web scraping. *Encyclopedia of big data*, 1, 2017.