ISSN 2616-5775 Vol. 6, Issue 9: 25-31, DOI: 10.25236/AJCIS.2023.060904

Design and Visualization of Python Web Scraping Based on Third-Party Libraries and Selenium Tools

Shujun Yuan

School of Information Engineering, Hubei University of Economics, Wuhan, China
3532415763@qq.com

Abstract: *The aim of this study is to analyze the data from Chinese movie websites to understand the trend distribution of movie genres and ratings. It used Python third-party libraries and the Selenium tool to crawl data from various movie websites and platforms. Douban Films is one of the most prominent applications. In order to realize the data analysis of Douban movies, the crawler program was designed from multiple perspectives, including two data capture channels, keyword search movies and screening search rankings. By viewing the movie details function module, it can achieve the requirements of obtaining movie ratings, stars, online viewing addresses, cloud disk search links and film and television download resources. Visualization of the data results was conducted using the third-party Python graph library Matplotlib. The results showed that the film rating and the total number of ratings are important factors that ordinary users refer to when watching films. Drama films are the most popular type of film among producers and film companies, while adventure films are the type of film that is easily overlooked by viewers. These data analyses can reflect the public's viewing trends under the guidance of consumers.*

Keywords: *Python; Web scrapy; Visualization; Selenium; Movie websites*

1. Introduction

In the era of big data and artificial intelligence, the public has come to realize that data acquisition is the primary task of data mining and analysis. With the rapid development of the Internet, the amount of data available online is becoming increasingly large. This data is a valuable resource for research, but it can be difficult and time-consuming to collect and process manually. By using Web scraping technology, we can obtain more authentic, comprehensive, and valuable data information. In the field of film studies, web crawlers have been used to collect data on a variety of topics, including movie ratings, trends, and reviews. This data can be used to track the popularity of different films, identify emerging trends, and understand the factors that influence filmgoers' choices. The analysis of film trends can help film companies understand the preferences of their target audiences. By analyzing data on movie ratings, film companies can identify which genres are popular and which are not. This information can be used to make decisions about what types of movies to produce and market. In this paper, the author used the Python standard library to study the principles of web crawling and implement data capture on the movie interface. He then complete the design of the web crawler program and finally perform data analysis and visualization on the acquired data.

After completing the data collection, the data cleaning and preprocessing are performed. In the visualization section, the WordCloud library is used in combination with the Matplotlib library to draw word clouds, pie charts, bar charts, and other model charts. Pie charts are used to analyze the rating levels. Text data is replaced by graphics, which can show the distribution of rating levels for movies by the public. Word cloud charts are used for analysis, and the data is displayed under frequency statistics, with key words highlighted more prominently.

Python crawler technology can be used to automate the process of data collection. The author was not satisfied with the information provided by existing film platforms, and wanted to be able to obtain information such as streaming links, cloud storage search links, and resource download links. Therefore, the study decided to design a film crawler program based on third-party libraries and the Selenium tool to help teachers and students obtain film data and analysis results.

It conducted a study on the data of domestic movie websites and analyzed the development of movies from the perspective of the relationship between movie genres and ratings. The data was visualized. By using crawling technology to obtain user ratings, the author analyzed the user's love for different movie

ISSN 2616-5775 Vol. 6, Issue 9: 25-31, DOI: 10.25236/AJCIS.2023.060904

genres. The analysis results were used to help users make decisions and help producers adjust their marketing strategies. Based on the data characteristics of the film website platform, the study crawled the required data from it, cleaned and preprocessed the crawled data, and performed visual analysis. The entire process was implemented in Python, among which:

- Data scraping used the Selenium tool to implement the WebDriver cross-platform, and the urlib3 library operated the web page url and processed the web page content.
- Data pre-processing was implemented using Numpy and Pandas modules.
- Data visualization analysis was implemented using Matplotlib library.
- A Graphical User Interface program was created using the User Interface framework Tkinter.

2. Environment setup and overall scheme design

Matplotlib is a plotting library in the Python standard library. It can draw various statistical graphs, such as scatter plots, bar charts, pie charts, etc., based on the array calculation functions of NumPy module. It is suitable for data visualization analysis. Matplotlib.Pyplot is a library that provides a comprehensive set of functions for creating static, animated, and interactive visualizations in Python. The Pyplot submodule provides the bar() and pie() functions to draw bar charts and pie charts, respectively. The show() function is used to display the current figure that is being processed. Seaborn is a statistical plotting library built on top of Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics. Seaborn includes a variety of built-in themes and styles that can be used to customize the appearance of Matplotlib plots. It also supports visualization of data structures from the NumPy and Pandas modules. In my study, the development environment system is Windows. With deploying the Python development environment, Anaconda was installed and implemented. Anaconda is a free and open-source distribution version that includes over 180 scientific packages and dependencies that are needed, including Conda and Python. It supports multiple platforms and provides package management and environment management functions, which can solve the headache problem of coexisting Python under multiple versions and installing various third-party packages.

XPath is a powerful tool for selecting nodes from XML files. It provides a concise and easy-to-use path expression syntax.This paper proposes a method for using XPath to extract data from movie websites. The method first uses the requests library to capture the HTML content of the web page. Then, it uses XPath positioning method to parse the desired nodes, such as the release date and title information of movies.

Selenium is a tool that uses a collection of open source APIs called WebDriver to automate testing of web applications. Using the Selenium web driver will reduce the time expended on testing. Selenium is a web based online testing tool[1]. The study used this tool to implement automated simulated login operations to establish data communication with domestic film platform websites.

The Multiprocessing.Dummy library imports the Pool module to implement multi-threaded crawling of different movie reviews with the same API. By using the Pool.map() function, the threads are executed sequentially to improve the running efficiency.

The Jieba library is based on the People's Daily corpus and returns a list variable of Chinese text after segmentation. It supports three segmentation modes: exact mode, full mode, and search engine mode. I mainly uses the lcut() function in the exact mode and imports the custom document to analyze the Chinese text of the comments.

The Pillow open-source library supports a variety of image formats, meeting the various needs of image processing in crawler programs. You do not need to install all supported external libraries to use Pillow’s basic features. Zlib and libjpeg are required by default[2]. Another feature is that the API design structure is simple and easy to get started, greatly reducing development difficulty. The study used this library to crawl the URL, download and save the image, and complete the image processing operation.

The implementation of the research project is mainly divided into three parts: environment configuration, visualization, and crawler program design. The flowchart of the crawler program can be seen in Figure 1. The first step is to complete the preparation by building the environment. Then, the data is captured and processed. The regular expression and Python function library are used to process the data and complete the cleaning. The relationship between the functional modules and the program is

ISSN 2616-5775 Vol. 6, Issue 9: 25-31, DOI: 10.25236/AJCIS.2023.060904

designed. The user interface is designed using the Tkinter framework to meet the needs and complete the interaction between the server and the client. Visualization analysis is based on the use of Python third-party libraries and open-source toolkits to display the results of data analysis.

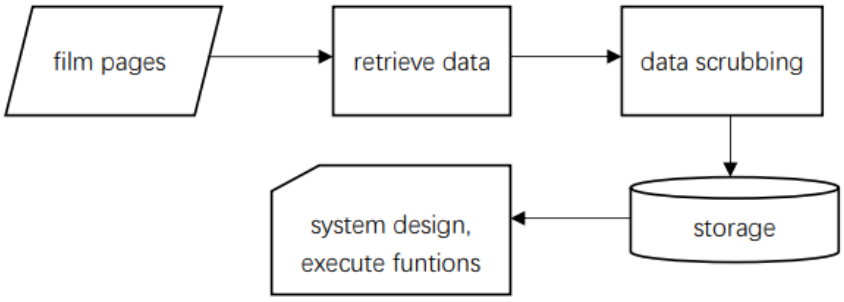

Figure 1: The flowchart of the crawler program.

3. Data acquisition and preprocessing

Most movie websites contain a wealth of movie information. However, the IP restrictions imposed on crawlers by these websites make it impossible to directly obtain all the data for the target movie. Therefore, I conducted Python crawler-related technologies to capture the data in a timely and efficient manner. Due to the anti-crawling mechanism of Douban, a movie website, I used the Selenium tool to simulate browser login when designing the crawler program. By parsing the source code of web page, I captured the required data through using the BeautifulSoup module and urlib3 library. The study used a try function to process the URL, disable the SSL certificate, and convert the JSON format to a Python object. The main source code is shown as follows:

```
    context = _create_unverified_context()
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36'}
url = 'https://movie.douban.com/j/chart/top_list?type=' + str(typeId) + '&interval_id=100:90&action=unwatched&start=0&limit=' + str(movie_count)
req = urllib.request.Request(url=url, headers=headers)
f = urllib.request.urlopen(req, context=context)
response = f.read()
jsonData = loads(response)
```

In the process of data crawling, it used a round-robin method to operate each target movie object with the following code:

```
for subData in jsonData:
    if (float(subData['rating'][0]) >= float(rating)) and (float(subData['vote_count']) >= float(vote_count)):
        sub_list= []
        sub_list.append(subData['title'])
        sub_list.append(subData['rating'][0])
        sub_list.append(subData['rank'])
        sub_list.append(subData['vote_count'])
        res_list.append(sub_list)
```

It is also important to pay attention to the handling of exceptions. The source code is as follows:

```
except Exception as ex:
    err_str = "confused error：  {}".format(ex)
    return [err_str]
except Exception as ex:
    load_driver_success = False
    err_str = "chromedriver false，  please download chromedriver and file in correct path.\n\n error message：  {}".format(ex)
    return [err_str]
    except Exception as ex:
        browser.quit()
```

ISSN 2616-5775 Vol. 6, Issue 9: 25-31, DOI: 10.25236/AJCIS.2023.060904

```
            err_str = "chromedriver running , but there is a bump in the environment : {}".format(ex)
            return [err_str]
```

To prevent the website that is being crawled from identifying the use of the Selenium tool to operate the WebDriver, the developer mode is set. At the same time, the access speed needs to be paid attention to and optimized. Next, for loading the Chrome driver, the driver status is detected, the execution path is set, the page loading and JavaScript loading are waited for, and the waiting timeout setting is completed. Finally, the data field information is collected by the get method, and the code is as follows:

```
chrome_options.add_experimental_option('excludeSwitches', ['enable-automation'])
chrome_options.add_experimental_option("prefs",{"profile.managed_default_content_settings.images": 2})
load_driver_success = False
browser = webdriver.Chrome(executable_path='D:/Program Files (x86)/chromedriver.exe', chrome_options=chrome_options)
browser.set_page_load_timeout(10)
browser.set_script_timeout(10)
wait = WebDriverWait(browser, 10)
browser.get('https://movie.douban.com/subject_search?search_text=' + urllib.parse.quote(key_word) + '&cat=1002')
```

In order to complete the work of capturing and processing web page content elements, Python urlib library is used to operate web page URL. The modules relationship of urlib package classes is shown in Figure 2.

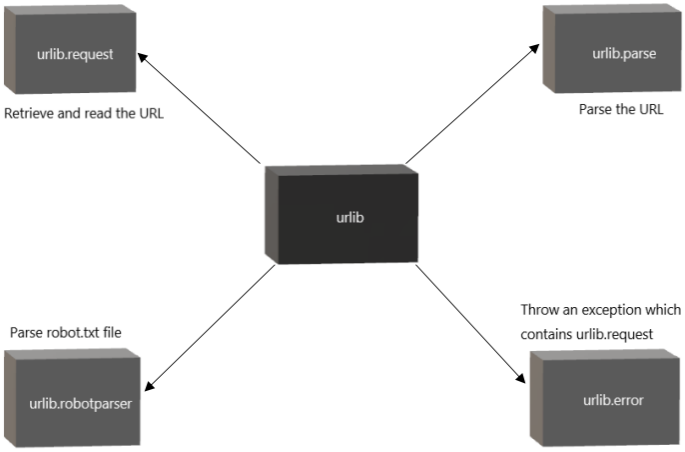

Figure 2: The modules relationship of urlib package classes.

In order to better analyze the distribution of popularity and ratings of different movie genres, it crawled the movie genres and rating data from the Douban box office rankings when obtaining data. By comparing the ratings of different genres, the study can better reflect the audience's preference for different types of movies. During the data crawling process, the web page structures of different websites are not the same. For the convenience of data visualization analysis, regular expressions and Python functions are used to preprocess the data, clean up useless information, and remove movies without ratings. The results of part of the data after collation are shown in Figure 3.

Search

genre docu count = 9 rating > 7.0 r_count > 10000 by ranking

f_name by keyword

name	rating	ranking	r_count
Puujee	9.7	1	48351
Twenty-Two	9.5	2	75256
RBG	9.5	3	77116
The Cove	9.3	4	358430
Village China	9.4	5	20162
Joe Hisaishi: The Music of Hayao Miyazaki	9.7	6	14332
My Love, Don't Cross That River	9.3	7	60398
The Rocking Sky	9.4	8	15534
My Octopus Teacher	9.3	9	43900

Figure 3: The results of part of the data after collation.

ISSN 2616-5775 Vol. 6, Issue 9: 25-31, DOI: 10.25236/AJCIS.2023.060904

4. Data Analysis and Visualization

4.1. Analysis foundation

With the rapid development of the film industry, the competition in the film market is becoming increasingly fierce. More and more film companies want to know the degree of public preference by analyzing the fluctuations of end-user review for different movie genres. On the one hand, the rating factor is a review and feedback of the film from the perspective of acceptance by the audience. It depends on the artistic quality of the film itself. On the other hand, it depends on the needs of the audience themselves, that is, to what extent the film is integrated with the content that the audience expects to see. Therefore, it is important to understand the impact of film types, ratings, and other factors on audience choice. First, the hot genres are analyzed from the number of film types. Second, the degree of audience acceptance of different film types is analyzed from the perspective of average rating. Then, the distribution of ratings is used to more realistically understand the satisfaction of most audiences with the film. Finally, the relationship between the number of comments and ratings is combined to further analyze customer sentiment towards the film.

4.2. Analyzing the degree of preference by cutting into the number of movie genres

In recent years, there have been 10 major film genres with more than 25 releases each. These genres include drama, comedy, romance, action, adventure, animation, suspense, thriller, science fiction, and crime. As shown in Figure 4, science fiction films have accounted for 32.25% of all released films, which is nearly twice the share of action films. This suggests that science fiction films are a popular choice for producers. Action and animation films ranked second and third in terms of total market share, respectively, and together these three genres accounted for 64.63% of the total market. As shown in the pie chart of film type quantity ratio, drama, romance, and comedy genres account for 35.37% of the total number, indicating that other film genres are still relatively abundant, but not many producers have chosen them.

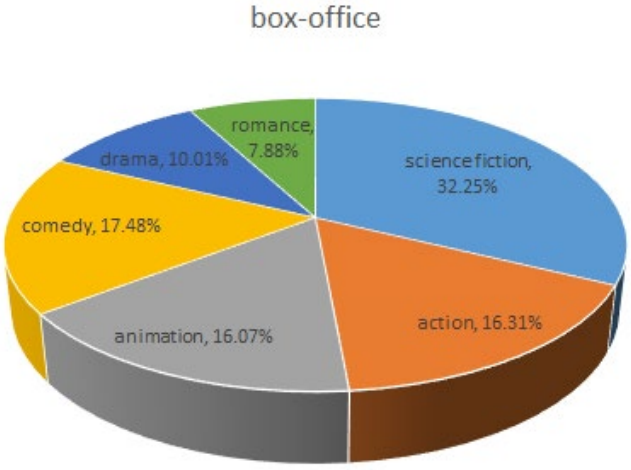

Figure 4: The pie chart of film type quantity ratio.

4.3. Analyzing the audience's love for different movie genres based on average ratings

An analysis of the average rating of film types in Figure 5 reveals that crime, action, drama, comedy, and adventure films all received ratings above 5.0. Among these genres, adventure films received the highest rating, with a strong emotional interaction with audiences. Crime and fantasy films, which are ranked second and third, respectively, also received high ratings, suggesting that films with emotional stimulation are appealing to audiences. In contrast, thriller films, which ranked last, received an average rating of only 4.0. This type of film has a relatively restricted audience and requires higher quality to be accepted by the general public.

ISSN 2616-5775 Vol. 6, Issue 9: 25-31, DOI: 10.25236/AJCIS.2023.060904

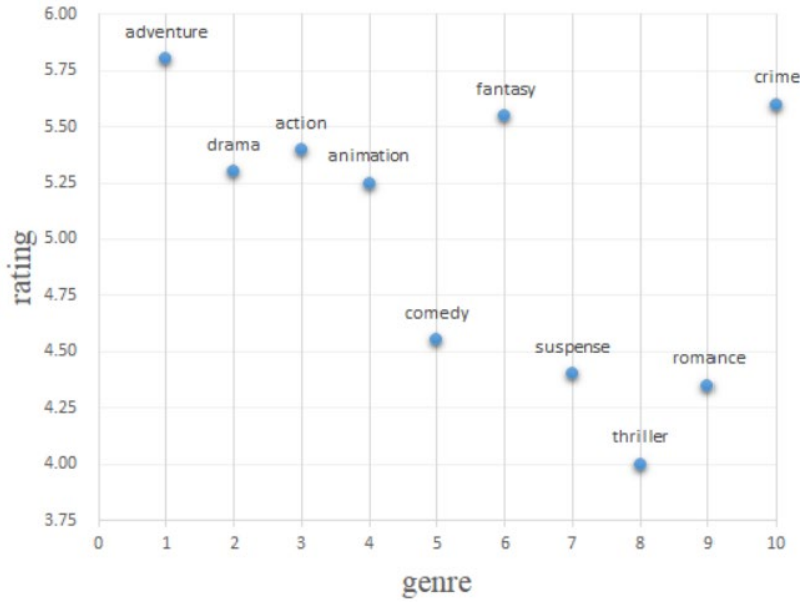

Figure 5: The scatter plot of film type average rating.

5. Crawler program design

5.1. Functional interaction interface design

In the process of image storage, the images are downloaded from the specified URLs and processed by the Pillow library. Python functions need to be packaged into threads for processing, and the adapter for the time processing function needs to be set separately. After the captured data is cleaned and stored in the database, it can be searched by field. For the user interface, the end-user experience should be considered, and the beauty of the interface setting should be the goal of human-computer interaction. The package class functions and parameters are defined as follows:

```
def thread_it(func, *args):
def handlerAdaptor(fun, **kwds):
def save_img(img_url, file_name, file_path):
def resize(w_box, h_box, pil_image):
def get_mid_str(content, startStr, endStr):
```

5.2. Graphics User Interface implementation

To meet the requirements of designing a graphical user interface (GUI) program, it used the Tkinter framework to handle the UI process. Then, the study designed the functions, including displaying the IMDb rating and response prompt on the display interface, and implementing the search function by binding events. The source code is shown below:

```
        def ui_process(self):
            root = Tk()
            self.root = root
            root.title("movie page assistant(filterable and downloadable custom movies)")
            self.center_window(root, 1000, 565)
            root.resizable(0, 0)
    # Frame set
            frame_root = Frame(labelframe, width=400)
            frame_l = Frame(frame_root)
            frame_r = Frame(frame_root)
            self.frame_root = frame_root
            self.frame_l = frame_l
            self.frame_r = frame_r
    # Event binding
```

ISSN 2616-5775 Vol. 6, Issue 9: 25-31, DOI: 10.25236/AJCIS.2023.060904

```
treeview.bind('<<TreeviewSelect>>', self.show_movie_data)
B_0.configure(command=lambda:thread_it(self.searh_movie_in_rating))
T_vote_keyword.bind('<Return>', handlerAdaptor(self.keyboard_T_vote_keyword))
project_statement.bind('<Enter>', self.project_statement_get_focus)
```

6. Conclusion & Discussion

This paper has demonstrated the application of the Python programming language in data crawling and visualization analysis, as well as its role in crawler program design. By analyzing the publicly released films on websites such as Douban and China Box Office, it is possible to convey some feedback to production companies, as well as provide important reference standards for users to watch movies. Based on the examples, using Python for visualization analysis can be used to apply a large amount of complex data, making the results more vivid and understandable to the public, and improving the completion of the work. In terms of program design, the application of multithreading technology can realize asynchronous execution of multiple keyword queries. The GUI can be added with frames to parallelize keyword search and ranking filtering. A multimodal interface with text and images is implemented, and by clicking on the button, you can view the details of the film and resource address. In addition, this project provides a glimpse into the role of Python APIs in the OpenMachineLearning field. OpenML-Python, a client API for Python, which opens up the OpenML platform for a wide range of Python-based machine learning tools. It provides easy access to all datasets, tasks and experiments on OpenML from within Python[3]. In the end, I intend to continue developing my technology stack in the direction of Python API applications and explore this area in future research.

References

[1] Nyamathulla S, Ratnababu P, Shaik N S. A Review on Selenium Web Driver with Python[J]. Annals of the Romanian Society for Cell Biology, 2021: 16760-16768.
[2] Clark A. Pillow (pil fork) documentation[J]. Readthedocs, 2015.
[3] Feurer M, Van Rijn J N, Kadra A, et al. Openml-python: an extensible python api for openml [J]. The Journal of Machine Learning Research, 2021, 22(1): 4573-4577.