ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 7, Issue 7, July 2018

Web Scraper Bot
to Harvest IMDB Data in Realtime

Nithin RS[1], Ankush Reddy P[2], Charan HN[3], Tejas KS[4], Sriranganath SM[5]

BE, Department of ECE, RNSIT, Bangalore, India[1]

BE, Department of TCE, Sir MVIT, Bangalore, India[2]

BE, Department of ECE, BIT, Bangalore, India[3]

BE, Department of CSE, EWIT, Bangalore, India[4]

BE, Department of CSE, JSSATE, Bangalore, India[5]

Abstract: Web mining is a process in which algorithms are written to analyse or discover patterns from the World Wide Web. Web mining may include web content mining, web structure mining or web usage mining. In this paper we have put in efforts to extract/mine data from the IMDb website-a leading movie website guide for watching movies, listening to music, watching TV shows, celebrity gossips and much more. We have written an algorithm/used a web scraper Bot by virtue of which the database of a particular year or name or IMDb rating is extracted in few seconds and is displayed in the CSV format. Various calculations and analysis can be further carried out after extraction of crude data from the website and converted to useful format. Efforts are also made to store the data in both processed and unprocessed format for future applications.

Keywords: Analyse or discover patterns from the World Wide Web, extract/mine data from the IMDb website, crude data from the website and converted to useful format, CSV format, store the data in both processed and unprocessed format

I. INTRODUCTION

Extraction of web content is very useful and is the trending field in the 21st century. [1] We employ web content mining, which is a process of web mining. [2] Few of the applications of this web scraper bot are

- Identify the topics represented by the web document.
- Categorize web documents.
- Applications related to relevance which includes use of filters, recommendation and/or task based relevance.

II. PROBLEM STATEMENT AND POSSIBLE SOLUTION

A. Problem

- To extract the web data from the IMDb website [3] based on the year of release, IMDb rating, budget and run time. Approximately 100s of movies get released each year and individual scrutiny of the above is a tedious task.
- Arrangement of the movies according to the rating and category would be beneficial for the user to categorically watch the movie.

B. Solution: A web scraper Bot

- An intelligent way of handling things is very much required in this space. An algorithm is the need of the day.
- Data can be extracted and stored in .csv format within minutes.
- A code is written in python to carry the above task.

III. METHODOLOGY

The IMDb website consists of tons of data regarding to movies and entertainment. Figure 1 shows the typical layout of one of the webpage of the IMDb website. IMDb website is a platform or guide to a movie-freak. The major challenge ahead is to extract the data from the IMDb website (crude data), analyse the patterns and draw conclusions. An algorithm is written in Python and various steps are followed.

- The URL of the webpage is mentioned as shown in figure 2 in the code.
- Figure 3 shows the ‘page.status code’ being checked for and figure 4 shows different status codes of 2XX success.
- Now the results are printed using suitable print statements as shown in figure 5 and 6.

DOI 10.17148/IJARCCE.2018.7717

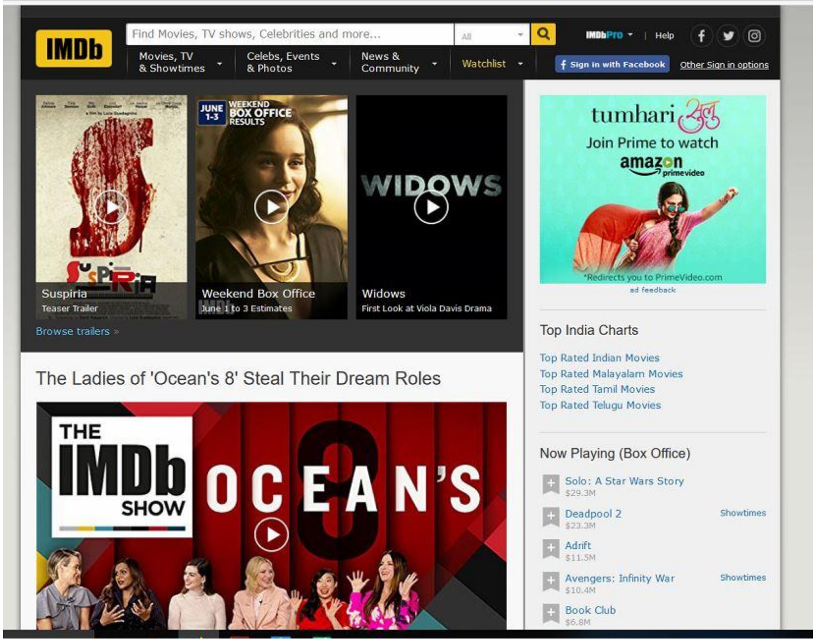
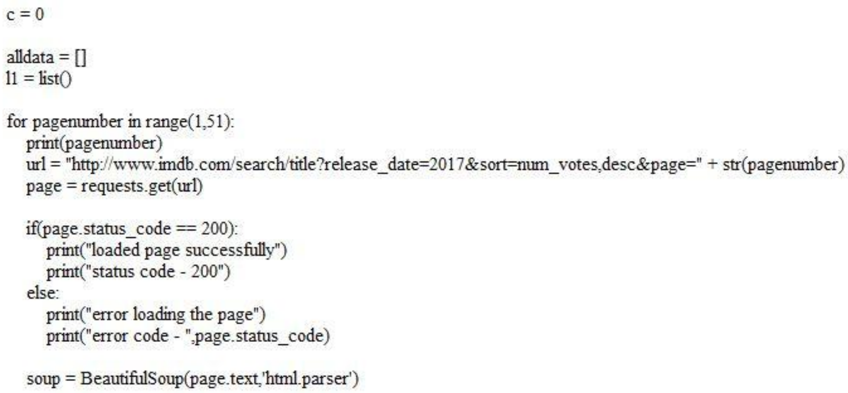
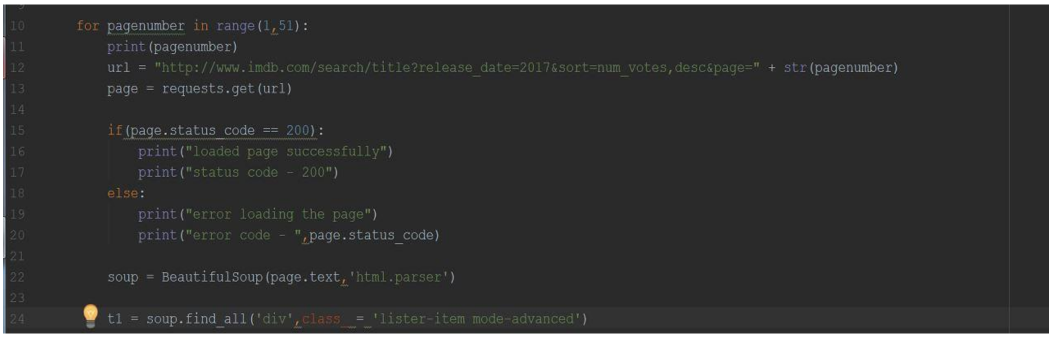


Figure 1 shows the typical layout of one of the webpage of the IMDb website.



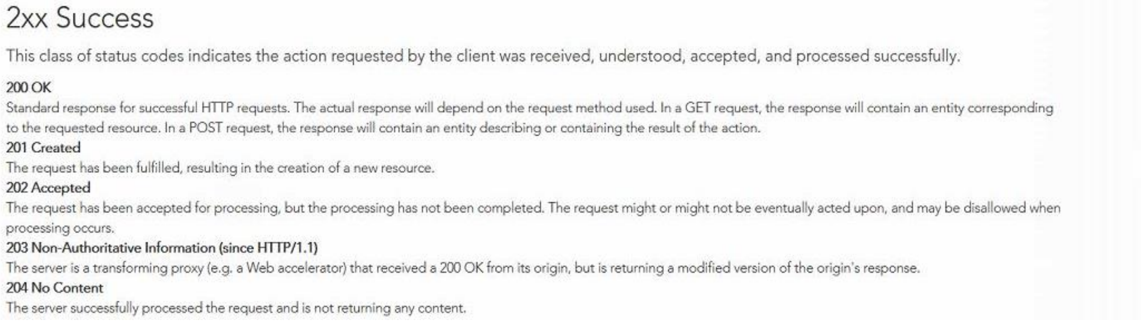
```
c = 0

alldata = []
l1 = list()

for pagenumber in range(1,51):
    print(pagenumber)
    url = "http://www.imdb.com/search/title?release_date=2017&sort=num_votes,desc&page=" + str(pagenumber)
    page = requests.get(url)

    if(page.status_code == 200):
        print("loaded page successfully")
        print("status code - 200")
    else:
        print("error loading the page")
        print("error code - ",page.status_code)

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

Figure 2 shows the URL of the website to be mined



```
    for pagenumber in range(1,51):
        print(pagenumber)
        url = "http://www.imdb.com/search/title?release_date=2017&sort=num_votes,desc&page=" + str(pagenumber)
        page = requests.get(url)

        if(page.status_code == 200):
            print("loaded page successfully")
            print("status code - 200")
        else:
            print("error loading the page")
            print("error code - ",page.status_code)

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

        t1 = soup.find_all('div',class_ = 'lister-item mode-advanced')
```

Figure 3 shows the ‘page. Status code’ being checked for



2xx Success

This class of status codes indicates the action requested by the client was received, understood, accepted, and processed successfully.

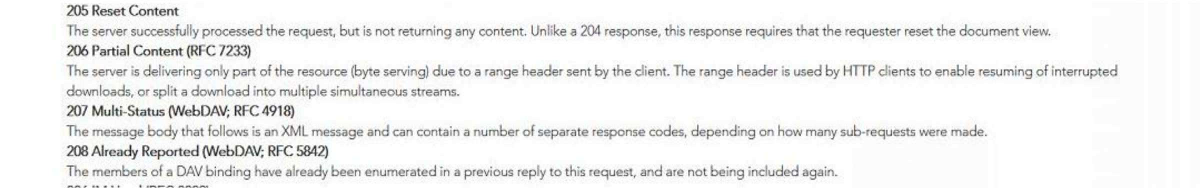
200 OK
Standard response for successful HTTP requests. The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource. In a POST request, the response will contain an entity describing or containing the result of the action.

201 Created
The request has been fulfilled, resulting in the creation of a new resource.

202 Accepted
The request has been accepted for processing, but the processing has not been completed. The request might or might not be eventually acted upon, and may be disallowed when processing occurs.

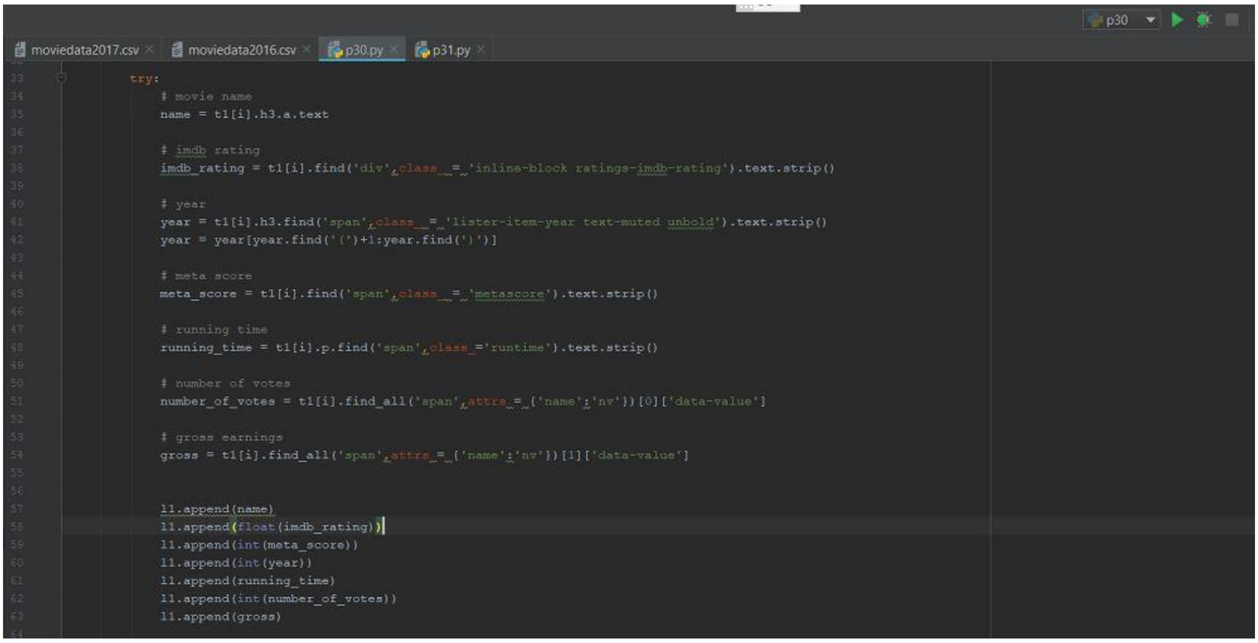
203 Non-Authoritative Information (since HTTP/1.1)
The server is a transforming proxy (e.g. a Web accelerator) that received a 200 OK from its origin, but is returning a modified version of the origin's response.

204 No Content
The server successfully processed the request and is not returning any content.



205 Reset Content
The server successfully processed the request, but is not returning any content. Unlike a 204 response, this response requires that the requester reset the document view.
206 Partial Content (RFC 7233)
The server is delivering only part of the resource (byte serving) due to a range header sent by the client. The range header is used by HTTP clients to enable resuming of interrupted downloads, or split a download into multiple simultaneous streams.
207 Multi-Status (WebDAV; RFC 4918)
The message body that follows is an XML message and can contain a number of separate response codes, depending on how many sub-requests were made.
208 Already Reported (WebDAV; RFC 5842)
The members of a DAV binding have already been enumerated in a previous reply to this request, and are not being included again.

Figure 4 shows different status codes of 2XX success.

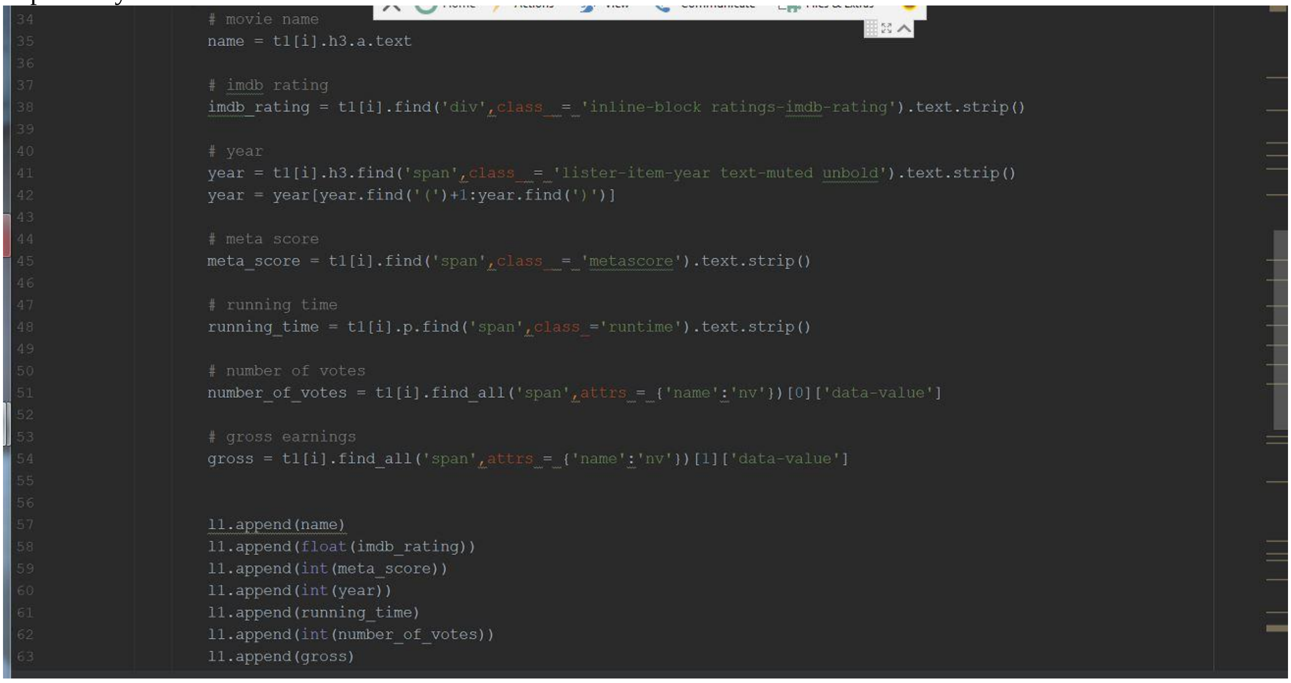


```
        try:
            # movie name
            name = t1[i].h3.a.text

            # imdb rating
            imdb_rating = t1[i].find('div',class_ = 'inline-block ratings-imdb-rating').text.strip()

            # year
            year = t1[i].h3.find('span',class_ = 'lister-item-year text-muted unbold').text.strip()
            year = year[year.find('(')+1:year.find(')')]

            # meta score
            meta_score = t1[i].find('span',class_ = 'metascore').text.strip()

            # running time
            running_time = t1[i].p.find('span',class_='runtime').text.strip()

            # number of votes
            number_of_votes = t1[i].find_all('span',attrs = {'name':'nv'})[0]['data-value']

            # gross earnings
            gross = t1[i].find_all('span',attrs = {'name':'nv'})[1]['data-value']


            l1.append(name)
            l1.append(float(imdb_rating))
            l1.append(int(meta_score))
            l1.append(int(year))
            l1.append(running_time)
            l1.append(int(number_of_votes))
            l1.append(gross)
```

Figure 5 shows the attributes to be printed as mentioned in the code

IV. RESULTS

The web scraper bot was used in web mining and the result was obtained in few seconds. Figure 7 shows the run window and the runtime result. The data after mining was analysed and orderly presented in .csv format. Figure 8 and 9 shows the data being mined from the IMDb website and includes movie details released in the year 2016 and 2017 respectively.



```
            # movie name
            name = t1[i].h3.a.text

            # imdb rating
            imdb_rating = t1[i].find('div',class_ = 'inline-block ratings-imdb-rating').text.strip()

            # year
            year = t1[i].h3.find('span',class_ = 'lister-item-year text-muted unbold').text.strip()
            year = year[year.find('(')+1:year.find(')')]

            # meta score
            meta_score = t1[i].find('span',class_ = 'metascore').text.strip()

            # running time
            running_time = t1[i].p.find('span',class_='runtime').text.strip()

            # number of votes
            number_of_votes = t1[i].find_all('span',attrs = {'name':'nv'})[0]['data-value']

            # gross earnings
            gross = t1[i].find_all('span',attrs = {'name':'nv'})[1]['data-value']


            l1.append(name)
            l1.append(float(imdb_rating))
            l1.append(int(meta_score))
            l1.append(int(year))
            l1.append(running_time)
            l1.append(int(number_of_votes))
            l1.append(gross)
```

Figure 6 shows the attributes to be printed as mentioned in the code

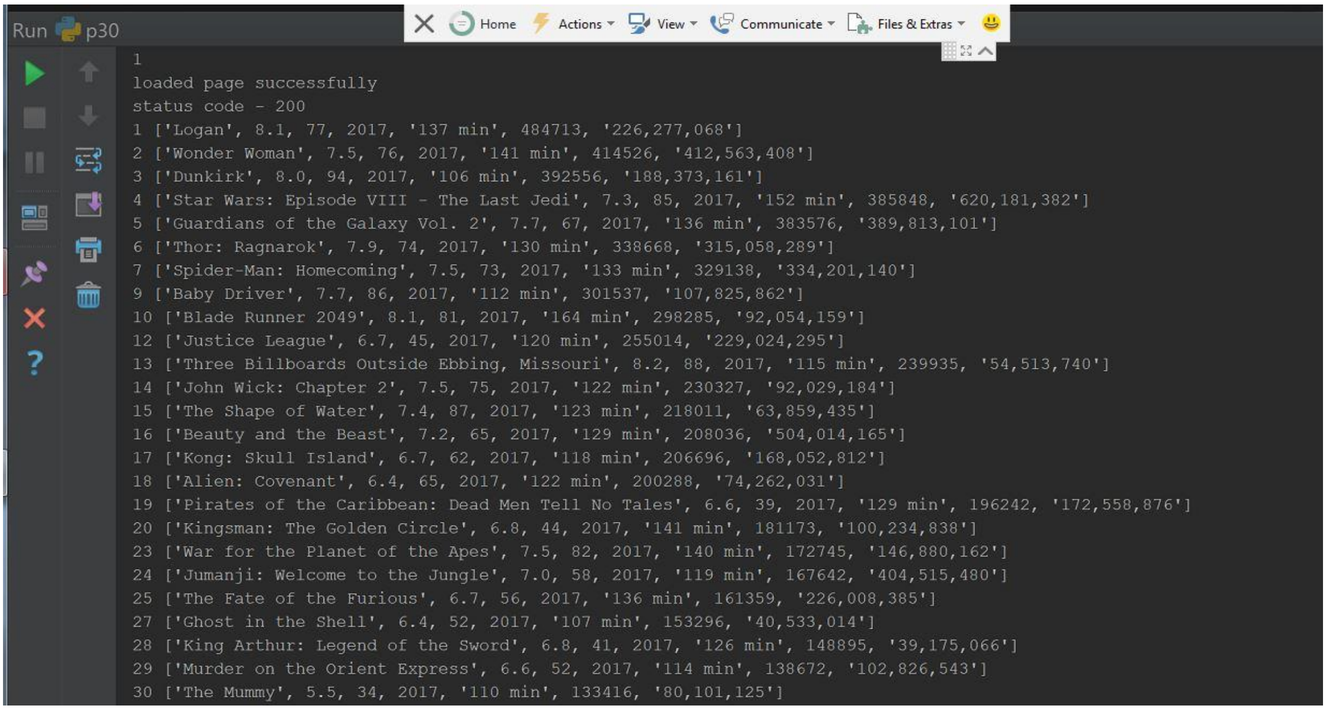
IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 7, Issue 7, July 2018

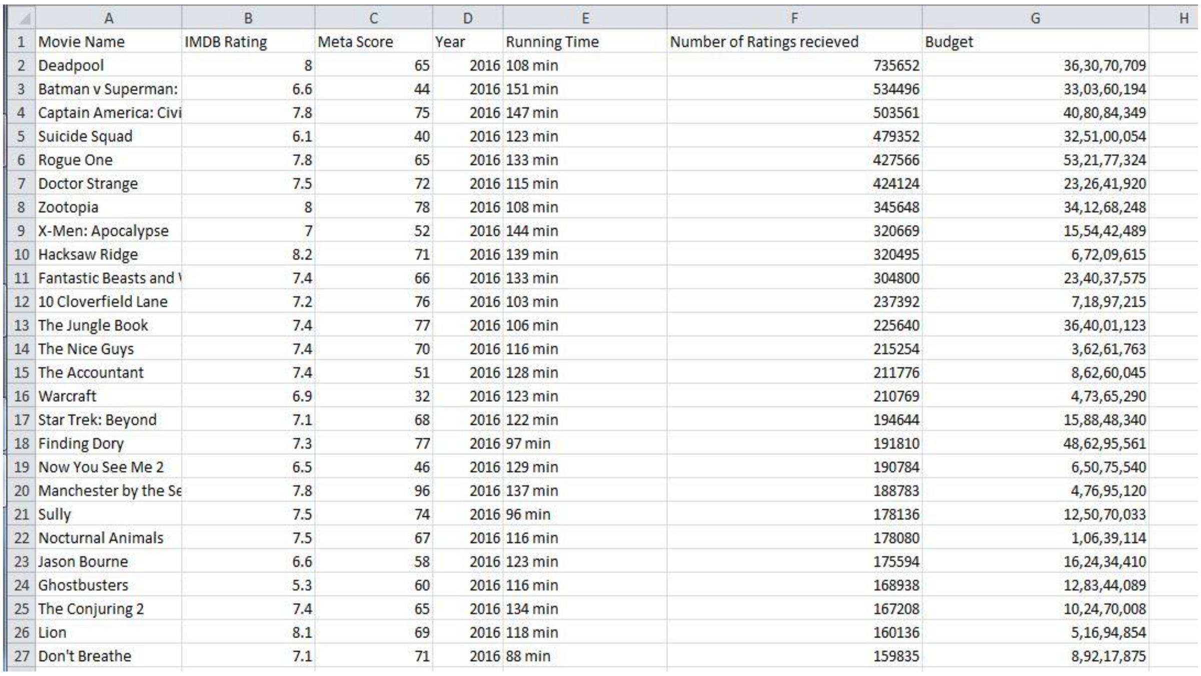


```
1
loaded page successfully
status code - 200
1 ['Logan', 8.1, 77, 2017, '137 min', 484713, '226,277,068']
2 ['Wonder Woman', 7.5, 76, 2017, '141 min', 414526, '412,563,408']
3 ['Dunkirk', 8.0, 94, 2017, '106 min', 392556, '188,373,161']
4 ['Star Wars: Episode VIII - The Last Jedi', 7.3, 85, 2017, '152 min', 385848, '620,181,382']
5 ['Guardians of the Galaxy Vol. 2', 7.7, 67, 2017, '136 min', 383576, '389,813,101']
6 ['Thor: Ragnarok', 7.9, 74, 2017, '130 min', 338668, '315,058,289']
7 ['Spider-Man: Homecoming', 7.5, 73, 2017, '133 min', 329138, '334,201,140']
9 ['Baby Driver', 7.7, 86, 2017, '112 min', 301537, '107,825,862']
10 ['Blade Runner 2049', 8.1, 81, 2017, '164 min', 298285, '92,054,159']
12 ['Justice League', 6.7, 45, 2017, '120 min', 255014, '229,024,295']
13 ['Three Billboards Outside Ebbing, Missouri', 8.2, 88, 2017, '115 min', 239935, '54,513,740']
14 ['John Wick: Chapter 2', 7.5, 75, 2017, '122 min', 230327, '92,029,184']
15 ['The Shape of Water', 7.4, 87, 2017, '123 min', 218011, '63,859,435']
16 ['Beauty and the Beast', 7.2, 65, 2017, '129 min', 208036, '504,014,165']
17 ['Kong: Skull Island', 6.7, 62, 2017, '118 min', 206696, '168,052,812']
18 ['Alien: Covenant', 6.4, 65, 2017, '122 min', 200288, '74,262,031']
19 ['Pirates of the Caribbean: Dead Men Tell No Tales', 6.6, 39, 2017, '129 min', 196242, '172,558,876']
20 ['Kingsman: The Golden Circle', 6.8, 44, 2017, '141 min', 181173, '100,234,838']
23 ['War for the Planet of the Apes', 7.5, 82, 2017, '140 min', 172745, '146,880,162']
24 ['Jumanji: Welcome to the Jungle', 7.0, 58, 2017, '119 min', 167642, '404,515,480']
25 ['The Fate of the Furious', 6.7, 56, 2017, '136 min', 161359, '226,008,385']
27 ['Ghost in the Shell', 6.4, 52, 2017, '107 min', 153296, '40,533,014']
28 ['King Arthur: Legend of the Sword', 6.8, 41, 2017, '126 min', 148895, '39,175,066']
29 ['Murder on the Orient Express', 6.6, 52, 2017, '114 min', 138672, '102,826,543']
30 ['The Mummy', 5.5, 34, 2017, '110 min', 133416, '80,101,125']
```

Figure 7 shows the run window after a successful run of the code

V. CONCLUSIONS

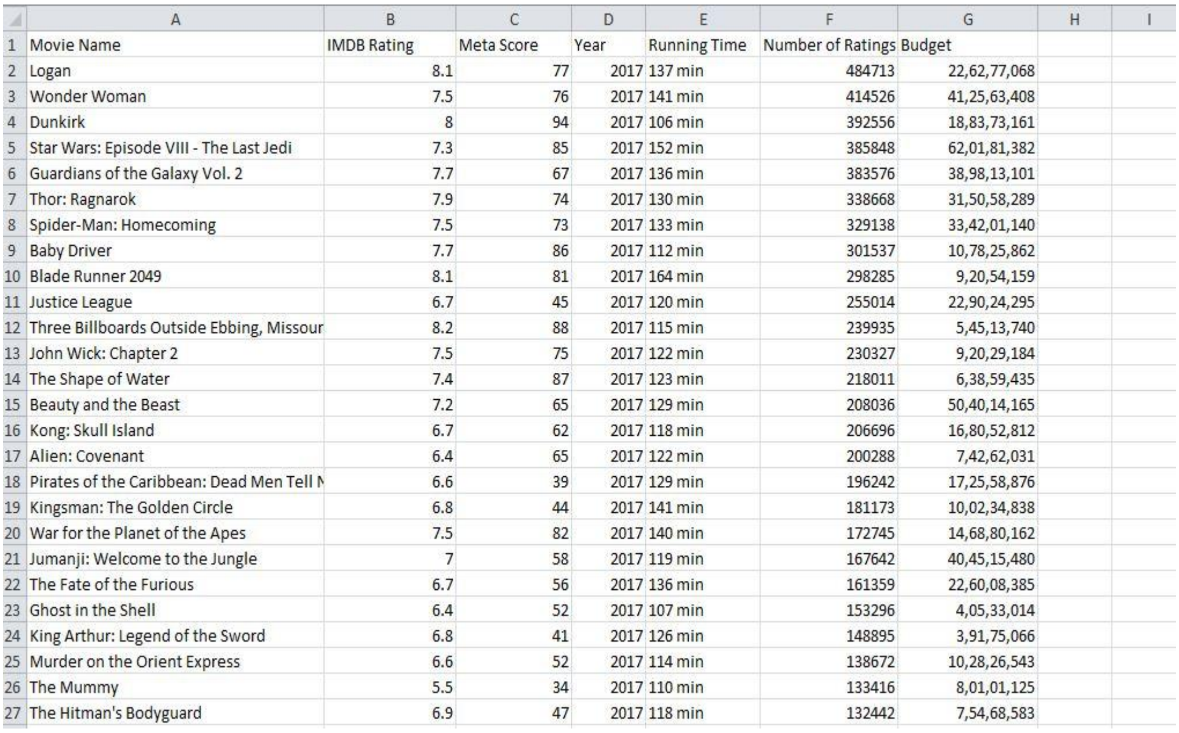
In this paper we have developed an 'IMDb Scraper Bot'- an intelligent way of extracting the contents of the website within few seconds with great accuracy. The code can be reused several number of times and may also be altered to suit the desired/ intended application. We have successfully mined the data and tabulated it in .csv format. All movies released during the year 2016 and 2017 have been tabulated with their rating, year of release, runtime, budget etc.



	A	B	C	D	E	F	G	H
1	Movie Name	IMDB Rating	Meta Score	Year	Running Time	Number of Ratings recieved	Budget	
2	Deadpool	8	65	2016	108 min	735652	36,30,70,709	
3	Batman v Superman:	6.6	44	2016	151 min	534496	33,03,60,194	
4	Captain America: Civi	7.8	75	2016	147 min	503561	40,80,84,349	
5	Suicide Squad	6.1	40	2016	123 min	479352	32,51,00,054	
6	Rogue One	7.8	65	2016	133 min	427566	53,21,77,324	
7	Doctor Strange	7.5	72	2016	115 min	424124	23,26,41,920	
8	Zootopia	8	78	2016	108 min	345648	34,12,68,248	
9	X-Men: Apocalypse	7	52	2016	144 min	320669	15,54,42,489	
10	Hacksaw Ridge	8.2	71	2016	139 min	320495	6,72,09,615	
11	Fantastic Beasts and \	7.4	66	2016	133 min	304800	23,40,37,575	
12	10 Cloverfield Lane	7.2	76	2016	103 min	237392	7,18,97,215	
13	The Jungle Book	7.4	77	2016	106 min	225640	36,40,01,123	
14	The Nice Guys	7.4	70	2016	116 min	215254	3,62,61,763	
15	The Accountant	7.4	51	2016	128 min	211776	8,62,60,045	
16	Warcraft	6.9	32	2016	123 min	210769	4,73,65,290	
17	Star Trek: Beyond	7.1	68	2016	122 min	194644	15,88,48,340	
18	Finding Dory	7.3	77	2016	97 min	191810	48,62,95,561	
19	Now You See Me 2	6.5	46	2016	129 min	190784	6,50,75,540	
20	Manchester by the Se	7.8	96	2016	137 min	188783	4,76,95,120	
21	Sully	7.5	74	2016	96 min	178136	12,50,70,033	
22	Nocturnal Animals	7.5	67	2016	116 min	178080	1,06,39,114	
23	Jason Bourne	6.6	58	2016	123 min	175594	16,24,34,410	
24	Ghostbusters	5.3	60	2016	116 min	168938	12,83,44,089	
25	The Conjuring 2	7.4	65	2016	134 min	167208	10,24,70,008	
26	Lion	8.1	69	2016	118 min	160136	5,16,94,854	
27	Don't Breathe	7.1	71	2016	88 min	159835	8,92,17,875	

Figure 8 shows all the movies released during the year 2016

DOI 10.17148/IJARCCE.2018.7717



	A	B	C	D	E	F	G	H	I
1	Movie Name	IMDB Rating	Meta Score	Year	Running Time	Number of Ratings	Budget		
2	Logan	8.1	77	2017	137 min	484713	22,62,77,068		
3	Wonder Woman	7.5	76	2017	141 min	414526	41,25,63,408		
4	Dunkirk	8	94	2017	106 min	392556	18,83,73,161		
5	Star Wars: Episode VIII - The Last Jedi	7.3	85	2017	152 min	385848	62,01,81,382		
6	Guardians of the Galaxy Vol. 2	7.7	67	2017	136 min	383576	38,98,13,101		
7	Thor: Ragnarok	7.9	74	2017	130 min	338668	31,50,58,289		
8	Spider-Man: Homecoming	7.5	73	2017	133 min	329138	33,42,01,140		
9	Baby Driver	7.7	86	2017	112 min	301537	10,78,25,862		
10	Blade Runner 2049	8.1	81	2017	164 min	298285	9,20,54,159		
11	Justice League	6.7	45	2017	120 min	255014	22,90,24,295		
12	Three Billboards Outside Ebbing, Missour	8.2	88	2017	115 min	239935	5,45,13,740		
13	John Wick: Chapter 2	7.5	75	2017	122 min	230327	9,20,29,184		
14	The Shape of Water	7.4	87	2017	123 min	218011	6,38,59,435		
15	Beauty and the Beast	7.2	65	2017	129 min	208036	50,40,14,165		
16	Kong: Skull Island	6.7	62	2017	118 min	206696	16,80,52,812		
17	Alien: Covenant	6.4	65	2017	122 min	200288	7,42,62,031		
18	Pirates of the Caribbean: Dead Men Tell N	6.6	39	2017	129 min	196242	17,25,58,876		
19	Kingsman: The Golden Circle	6.8	44	2017	141 min	181173	10,02,34,838		
20	War for the Planet of the Apes	7.5	82	2017	140 min	172745	14,68,80,162		
21	Jumanji: Welcome to the Jungle	7	58	2017	119 min	167642	40,45,15,480		
22	The Fate of the Furious	6.7	56	2017	136 min	161359	22,60,08,385		
23	Ghost in the Shell	6.4	52	2017	107 min	153296	4,05,33,014		
24	King Arthur: Legend of the Sword	6.8	41	2017	126 min	148895	3,91,75,066		
25	Murder on the Orient Express	6.6	52	2017	114 min	138672	10,28,26,543		
26	The Mummy	5.5	34	2017	110 min	133416	8,01,01,125		
27	The Hitman's Bodyguard	6.9	47	2017	118 min	132442	7,54,68,583		

Figure 9 shows all the movies released during the year 2016

REFERENCES

[1]. Data Mining, Internet Marketing and Web Mining - ijarcce https://www.ijarcce.com/upload/2017/march-17/IJARCCE%20117.pdf
[2]. Web Mining - Data Analysis and Management Research Group dmr.cs.umn.edu/Papers/P2004_4.pdf
[3]. IMDb website-https://www.imdb.com/
[4]. Error Code List-https://www.errorcodelist.com/

BIOGRAPHY

VISHESH S born on 13th June 1992, hails from Bangalore (Karnataka) and has completed B.E in Telecommunication Engineering from VTU, Belgaum, Karnataka in 2015. He also worked as an intern under Dr. Shivananju BN, former Research Scholar, Department of Instrumentation, IISc, Bangalore. His research interests include Embedded Systems, Wireless Communication, BAN and Medical Electronics. He is also the Founder and Managing Director of the corporate company Konigtronics Private Limited. He has guided over a hundred students/interns/professionals in their research work and projects. He is also the co-author of many International Research Papers. He is currently pursuing his MBA in e-Business and PG Diploma in International Business. Presently Konigtronics Private Limited has extended its services in the field of Software Engineering and Webpage Designing. Konigtronics also conducts technical and non-technical workshops on various topics.