Web_scraping_assignment_1

March 29, 2024

1 Web Scraping

2 Assignment 1

2.1 Rajib Dutta

2.2 duttarajib78@gmail.com

2.3 Batch DS2402

[ ]:
```
import pandas as pd
import numpy as np
import requests as req
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
import re
import time
from bs4 import BeautifulSoup
from datetime import datetime
```

2.3.1 Write a python program to display IMDB's Top rated 100 Indian movies' data https://www.imdb.com/list/ls056092300/ (i.e. name, rating, year ofrelease) and make data frame.

[ ]:
```
def get_top_100_movies(url):
    html=req.get(url=url).content
    soup=BeautifulSoup(markup=html, from_encoding='lxml')
    movie_details=soup.findAll('div', class_='lister-item mode-detail')
    best_100_movies=pd.DataFrame(columns=['Name', 'Year', 'Runtime', 'Rating'])
    for i, movie_detail in enumerate(movie_details):
        name=movie_detail.find('div', class_='lister-item-content').a.text
        year_raw=movie_detail.find('div', class_='lister-item-content').
 ↪find('span', class_='lister-item-year text-muted unbold').text
        year_of_release=int(re.compile(pattern='\d{4}').search(year_raw).
 ↪group())
        runtime_in_min=int(re.compile(pattern='\d+').search(movie_detail.
 ↪find('span', class_='runtime').text).group())
```

```
        rating=float(movie_detail.find('span',␣
 ↪class_='ipl-rating-star__rating').text)
        best_100_movies.loc[i]=[name, year_of_release, runtime_in_min, rating]
    return best_100_movies.reset_index(drop=True)

url='https://www.imdb.com/list/ls056092300/'
get_top_100_movies(url)
```

[ ]:
```
                                  Name  Year  Runtime  Rating
0                       Ship of Theseus  2012      139     8.0
1                                Iruvar  1997      140     8.4
2                       Kaagaz Ke Phool  1959      148     7.8
3   Lagaan: Once Upon a Time in India    2001      224     8.1
4                       Pather Panchali  1955      125     8.2
..                                   …   …       …      …
95                          Apur Sansar  1959      105     8.4
96                          Kanchivaram  2008      117     8.2
97                       Monsoon Wedding  2001      114     7.3
98                                Black  2005      122     8.1
99                               Deewaar  1975      174     8.0

[100 rows x 4 columns]
```

2.3.2 Write a python program to scrape product name, price and discounts from https://peachmode.com/search?q=bags

[ ]:
```
def get_bag_details(url):
    edge_options=Options()
    edge_options.add_argument("--headless")
    driver=webdriver.Edge(edge_options)
    driver.get(url)
    time.sleep(5)
    html=driver.page_source
    soup=BeautifulSoup(html, 'lxml')
    product_details=soup.findAll('div', class_='st-product st-col-xs-6␣
 ↪st-col-sm-3 st-col-md-3')
    product_df=pd.DataFrame(columns=['Name', 'Price', 'Discount (%)'])
    for i, product_detail in enumerate(product_details):
        name=product_detail.find('div', class_='product-title').text.strip()
        price_before=float(product_detail.find('span', class_='price st-money␣
 ↪money done').text.replace(' ', '').strip().replace(',',''))
        price_after=float(product_detail.find('span', class_='discounted_price␣
 ↪st-money money done').text.replace(' ', '').strip().replace(',', ''))
        discount=round(((price_before-price_after)/price_before)*100, 2)
        product_df.loc[i]=[name, price_after, discount]
    return product_df.reset_index(drop=True)
```

```
url='https://peachmode.com/search?q=bags'
get_bag_details(url=url)
```

[ ]:

	Name	Price	Discount (%)
0	TMN - Women Green Stylish Vegan Leather Tricot…	549.0	50.05
1	TMN - Women Pink Vegan Leather Tote Bag	449.0	50.06
2	TMN - Women Mint Stylish Vegan Leather Tricote…	549.0	50.05
3	TMN - Women Cream Stylish Vegan Leather Tricot…	549.0	50.05
4	TMN - Women Black Vegan Leather Sling Bag with…	449.0	50.06
5	TMN - Women Beige Vegan Leather Golden Chain S…	449.0	50.06
6	TMN - Women Brown Vegan Leather Golden Chain S…	449.0	50.06
7	TMN - Women Pink Vegan Leather Golden Chain Sl…	449.0	50.06
8	TMN - Women Brown Vegan Leather Tote Bag	449.0	50.06
9	TMN - Women Black Vegan LeatherGolden Chain Sl…	449.0	50.06
10	TMN - Women Brown Vegan Leather Tote Bag	749.0	50.03
11	TMN - Women Green Vegan Leather Tote Bag	749.0	50.03
12	TMN - Women Black Vegan Leather Tote Bag	749.0	50.03
13	TMN - Women Black Vegan Leather Sling Bag	599.0	50.04
14	TMN - Women Pink Vegan Leather Sling Bag	549.0	50.05
15	TMN - Women Green Vegan Leather Sling Bag	549.0	50.05
16	TMN - Women Brown Vegan Leather Sling Bag	449.0	50.06
17	TMN - Women Brown Vegan Leather Sling Bag	449.0	50.06
18	TMN - Women Black Vegan Leather Sling Bag	699.0	50.04
19	TMN - Women Pink Vegan Leather Sling Bag	849.0	50.03

2.3.3 Write a python program to scrape cricket rankings from icc-cricket.com. You have to scrape:

- Top 10 ODI teams in men's cricket along with the records for matches, points and rating.
- Top 10 ODI Batsmen along with the records of their team and rating.
- Top 10 ODI bowlers along with the records of their team and rating.

[ ]:
```
def get_top_10_ODI_teams(url):
    edge_options=Options()
    edge_options.add_argument("--headless")
    driver=webdriver.Edge(edge_options)
    driver.get(url)
    time.sleep(5)
    html=driver.page_source
    soup=BeautifulSoup(html, 'lxml')
    team_table=soup.find('div', class_='si-table')
    rows=team_table.find('div', class_='si-table-body').findAll('div',␣
 ↪class_='si-table-row')
    teams_df=pd.DataFrame(columns=['Team', 'Matches', 'Points', 'Rating',␣
 ↪'Position'])
    for i, row in enumerate(rows):
        if i==10:
```

```
            break
        position=int(row.find('div', class_='si-table-data si-pos').text.
↪strip())
        team=row.find('div', class_='si-table-data si-team').find('span',␣
↪'si-fname si-text').text.strip()
        matches=int(row.find('div', class_='si-table-data si-matches').span.
↪text.strip())
        points=int(row.find('div', class_='si-table-data si-pts').span.text.
↪strip())
        rating=int(row.find('div', class_='si-table-data si-rating').span.text.
↪strip())
        teams_df.loc[i]=[team, matches, points, rating, position]
    return teams_df.reset_index(drop=True)

url='https://www.icc-cricket.com/rankings/team-rankings/mens/odi'
get_top_10_ODI_teams(url)
```

[ ]:

	Team	Matches	Points	Rating	Position
0	India	58	7020	121	1
1	Australia	45	5309	118	2
2	South Africa	37	4062	110	3
3	Pakistan	36	3922	109	4
4	New Zealand	46	4708	102	5
5	England	41	3934	96	6
6	Sri Lanka	55	4947	90	7
7	Bangladesh	50	4417	88	8
8	Afghanistan	35	2845	81	9
9	West Indies	44	3109	71	10

[ ]:
```
def get_top_10_ODI_matsman(url):
    edge_options=Options()
    edge_options.add_argument("--headless")
    driver=webdriver.Edge(edge_options)
    driver.get(url)
    time.sleep(5)
    html=driver.page_source
    soup=BeautifulSoup(html, 'lxml')
    batsman_table=soup.find('div', class_='si-table')
    rows=batsman_table.find('div', class_='si-table-body').findAll('div',␣
↪class_='si-table-row')
    batsman_df=pd.DataFrame(columns=['Team', 'Player', 'Rating', 'Career Best␣
↪Rating', 'Position'])
    for i, row in enumerate(rows):
        if i==10:
            break
        position=row.find('div', class_='si-table-data si-pos').text.strip()
```

```
        team=row.find('div', class_='si-table-data si-team').find('span',␣
↪'si-fname si-text').text.strip()
        fname=row.find('span', class_='si-text si-fname').text.strip()
        lname=row.find('span', class_='si-text si-lname').text.strip()
        if fname is None:
            fname=''
        if lname is None:
            lname=''
        name=f'{fname} {lname}'
        rating=int(row.find('div', class_='si-table-data si-rating').span.text.
↪strip())
        best_rating=row.find('div', class_='si-table-data si-best').span.text.
↪strip()
        batsman_df.loc[i]=[team, name, rating, best_rating, position]
    return batsman_df.reset_index(drop=True)

url='https://www.icc-cricket.com/rankings/batting/mens/odi'
get_top_10_ODI_matsman(url)
```

[ ]:

	Team	Player	Rating	\
0	Pakistan	Babar Azam	824	
1	India	Shubman Gill	801	
2	India	Virat Kohli	768	
3	India	Rohit Sharma	746	
4	Ireland	Harry Tector	746	
5	New Zealand	Daryl Mitchell	728	
6	Australia	David Warner	723	
7	Sri Lanka	Pathum Nissanka	711	
8	England	Dawid Malan	707	
9	South Africa	Rassie van der Dussen	701	

	Career Best Rating	Position
0	898 v West Indies at Multan 2022	01
1	847 v Australia at Indore 2023	02
2	909 v England at Headingley 2018	03
3	882 v Sri Lanka at Headingley 2019	04 1
4	767 v Afghanistan at Sharjah 2024	=
5	750 v India at Mumbai 2023	06
6	869 v Pakistan at Adelaide 2017	07
7	728 v Bangladesh at Chittagong 2024	08 3
8	730 v Netherlands at Pune 2023	09 1
9	796 v England at Durham 2022	10 1

[ ]:
```
def get_top_10_ODI_bowler(url):
    edge_options=Options()
    edge_options.add_argument("--headless")
    driver=webdriver.Edge(edge_options)
```

```
    driver.get(url)
    time.sleep(5)
    html=driver.page_source
    soup=BeautifulSoup(html, 'lxml')
    bowler_table=soup.find('div', class_='si-table')
    rows=bowler_table.find('div', class_='si-table-body').findAll('div',␣
 ↪class_='si-table-row')
    bowler_df=pd.DataFrame(columns=['Team', 'Player', 'Rating', 'Career Best␣
 ↪Rating', 'Position'])
    for i, row in enumerate(rows):
        if i==10:
            break
        position=row.find('div', class_='si-table-data si-pos').text.strip()
        team=row.find('div', class_='si-table-data si-team').find('span',␣
 ↪'si-fname si-text').text.strip()
        fname=row.find('span', class_='si-text si-fname').text.strip()
        lname=row.find('span', class_='si-text si-lname').text.strip()
        if fname is None:
            fname=''
        if lname is None:
            lname=''
        name=f'{fname} {lname}'
        rating=int(row.find('div', class_='si-table-data si-rating').span.text.
 ↪strip())
        best_rating=row.find('div', class_='si-table-data si-best').span.text.
 ↪strip()
        bowler_df.loc[i]=[team, name, rating, best_rating, position]
    return bowler_df.reset_index(drop=True)

url='https://www.icc-cricket.com/rankings/bowling/mens/odi'
get_top_10_ODI_bowler(url)
```

[ ]:

	Team	Player	Rating	Career Best Rating	Position
0	South Africa	Keshav Maharaj	716	741 v Australia at Kolkata 2023	01
1	Australia	Josh Hazlewood	688	727 v England at Melbourne 2022	02
2	Australia	Adam Zampa	686		
3	India	Mohammed Siraj	678		
4	India	Jasprit Bumrah	665		
5	Afghanistan	Mohammad Nabi	656		
6	Pakistan	Shaheen Afridi	650		
7	India	Kuldeep Yadav	645		
8	New Zealand	Trent Boult	643		
9	Namibia	Bernard Scholtz	642		

```
2              695 v Bangladesh at Pune 2023       03
3           736 v New Zealand at Raipur 2023       04
4       838 v West Indies at Trivandrum 2018       05
5              657 v Zimbabwe at Harare 2022     06 1
6           688 v West Indies at Multan 2022     07 1
7         760 v New Zealand at Bay Oval 2019       08
8             775 v Australia at Cairns 2022       09
9  642 v Netherlands at Kirtipur, Nepal 2024       10
```

2.3.4 Write a python program to scrape details of all the posts from https://www.patreon.com/coreyms .Scrape the heading, date, content and the likes for the video from the link for the youtube video from the post.

[ ]:
```
def get_post_details(url):
    edge_options=Options()
    edge_options.add_argument("--headless")
    driver=webdriver.Edge(edge_options)
    driver.get(url)
    time.sleep(5)
    html=driver.page_source
    soup=BeautifulSoup(html, 'lxml')
    posts=soup.findAll('div', class_='sc-jvvksu sc-edERGn jUuDJC irJoS')
    posts_df=pd.DataFrame(columns=['Heading', 'content', 'date', 'likes'])
    for i, post in enumerate(posts):
        heading=post.find('span', class_='sc-1cvoi1y-0 hxhWXn').text.strip()
        content=post.find('div', class_='sc-cfnzm4-0 kJujbw').text.strip()
        date=post.find('div', class_='sc-lgu5zg-0 dXpjXs').text.strip()
        likes=int(post.find('span', class_='sc-bqiRlB etUZPh').text.strip())
        posts_df.loc[i]=[heading, content, date, likes]
    return posts_df.reset_index(drop=True)

url='https://www.patreon.com/coreyms'
get_post_details(url=url)
```

[ ]:
```
                                          Heading  \
0   How to Use ChatGPT as a Powerful Tool for Prog…
1   Python Tutorial: Simulate the Powerball lotter…
2   (Early Access) Python YouTube API Tutorial: Ca…
3   Python YouTube API Tutorial: Getting Started -…
4   Python Tutorial: Real World Example - Using Pa…
5                              Patreon Perk Updates
6   Python Pandas Tutorial (Part 11): Reading/Writ…
7          Live Stream - Chat, Q&A, Brews, and Coding
8   Python Pandas Tutorial (Part 10): Working with…
9   Python Pandas Tutorial (Part 9): Cleaning Data…
10  Python Pandas Tutorial (Part 8): Grouping and …
11      Python Pandas Tutorial (Part 7): Sorting Data
```

```
12  Python Pandas Tutorial (Part 6): Add/Remove Ro…
13  Python Pandas Tutorial (Part 5): Updating Rows…
14  Python Pandas Tutorial (Part 4): Filtering - U…
15  Python Pandas Tutorial (Part 3): Indexes - How…
16  Python Pandas Tutorial (Part 2): DataFrame and…
17  Python Pandas Tutorial (Part 1): Getting Start…
18                        First Half of Pandas Series
19  Python Tutorial: Zip Files - Creating and Extr…
```

	content	date	likes
0		May 22, 2023	2
1		Jan 28, 2023	2
2		Jun 10, 2020	2
3	In this Python Programming Tutorial, we'll be …	May 30, 2020	5
4	In this Python Programming Tutorial, we'll be …	May 25, 2020	5
5	Hi Everyone!I really appreciate all of the sup…	May 10, 2020	6
6	In this Python Programming video, we will be l…	Apr 2, 2020	4
7	Thanks for joining me on the live stream the o…	Mar 26, 2020	6
8	In this Python Programming video, we will be l…	Mar 18, 2020	2
9	In this Python Programming video, we will be l…	Feb 25, 2020	1
10	In this Python Programming video, we will be l…	Feb 15, 2020	1
11	In this Python Programming video, we will be l…	Feb 15, 2020	1
12	In this Python Programming video, we will be l…	Feb 2, 2020	2
13	In this Python Programming video, we will be l…	Feb 2, 2020	2
14	In this Python Programming video, we will be l…	Jan 18, 2020	2
15	In this Python Programming video, we will be l…	Jan 18, 2020	2
16	In this Python Programming video, we will be l…	Jan 11, 2020	2
17	In this Python Programming video, we will be l…	Jan 10, 2020	3
18	Hey everyone. Happy 2020! I know a lot of you …	Jan 6, 2020	7
19	In this video, we will be learning how to crea…	Nov 20, 2019	1

2.3.5 Write a python program to scrape house details from mentioned URL. It should include house title, location, area, EMI and price from https://www.nobroker.in/ .Enter three localities which are Indira Nagar, Jayanagar, Rajaji Nagar.

[ ]:
```python
def get_house_details(url):
    edge_options=Options()
    edge_options.add_argument("--headless")
    driver=webdriver.Edge(edge_options)
    driver.get(url)
    time.sleep(5)
    html=driver.page_source
    soup=BeautifulSoup(html, 'lxml')
    house_details=soup.findAll('div', style='box-shadow:0px 2px 4px #00000029')
    house_df=pd.DataFrame(columns=['title', 'location', 'area', 'emi', 'price'])
    for i, house_detail in enumerate(house_details):
```

```
        title=house_detail.find('h2', class_='heading-6 flex items-center␣
 ↪font-semi-bold m-0').a.text.strip()
        location_area=house_detail.find('div', class_='mt-0.5p overflow-hidden␣
 ↪overflow-ellipsis whitespace-nowrap max-w-70 text-gray-light leading-4 po:
 ↪mb-0.1p po:max-w-95').text.strip()
        location1, location2=location_area.split('\xa0')[0], location_area.
 ↪split('\xa0')[-1]
        location=f'{location1}, {location2}'
        area=house_detail.find('div', id ='unitCode').text.strip()
        price=house_detail.find('div', id ='minDeposit').find('div',␣
 ↪class_='font-semi-bold heading-6').text.strip().replace(' ', '')
        emi=house_detail.find('div', id ='roomType').text.strip().replace(' ',␣
 ↪'')
        house_df.loc[i]=[title, location, area, emi, price]
    return house_df.reset_index(drop=True)

url='https://www.nobroker.in/property/sale/bangalore/multiple?
 ↪searchParam=W3sibGF0IjoxMi45NzgzNjkyLCJsb24iOjc3LjY0MDgzNTYsInBsYWNlSWQiOiJDaElKa1FOM0dLUVd
 ↪0&city=bangalore&locality=Indiranagar,Jayanagar,Rajaji%20Nagar%20Industrial%20Town'
get_house_details(url=url)
```

[ ]:

	title	location	area
0	4+ BHK House For Sale In Jayanagar	Independent House, Tilak Nagar, near S.K. Bake…	2,500 sqft
1	3 BHK Flat In Santa Clara Apartment For Sale …	3Rd Cross Rd18th main, 9th Block, Marenahalli,…	1,395 sqft
2	2 BHK Flat In Pallava Terrace Apartments For S…	17 north main yediyur behind samudaya bhavan b…	1,050 sqft
3	3 BHK Flat In Nilavu For Sale In Indiranagar	Indira Nagar II Stage, Hoysala Nagar, Near BDA…	1,600 sqft
4	3 BHK House For Sale In Jayanagar 9th Block	Independent House, 125a, 38th B Cross Road, ne…	1,400 sqft
5	3 BHK Apartment In Lodha Azur For Sale near Ja…		
6	2 BHK Flat In Shanthi Park Apartments For Sale…		
7	2 BHK Apartment In Gopalan Admirality Court, I…		
8	3 BHK House For Sale In Jayanagar		
9	3 BHK Apartment In Codename Indiranagar For Sa…		
10	2 BHK Flat In Jayanagar Residency For Sale In…		
11	3 BHK Flat In Sri Maruthi For Sale In Jayanagar		
12	3 BHK Apartment In Gokul Lake View Apartments …		
13	3 BHK Flat In Mayitri Enclave For Sale In Jay…		
14	4 BHK Apartment In B And B Opulent Spire For S…		
15	4+ BHK House For Sale In Indiranagar Metro St…		
16	3 BHK Flat In Greenview Aparment For Sale In…		
17	4+ BHK House For Sale In Jayanagar		

```
5                 Lodha Azur,  Akshay Nagar,Bangalore  1,710 sqft
6   Shanthi Park Apartments, Kottapalya, Jayanagar…  1,100 sqft
7   Gopalan Admirality Court, Indiranagar,   6th M…  1,360 sqft
8   Independent House,  9th Block,Near  MEWA - VAN…  1,200 sqft
9      Codename Indiranagar,  Indiranagar, Bangalore.  2,000 sqft
10  jayanager residency. near D mart, Jayanager 3r…  1,167 sqft
11  Indira Gandhi Institute Of Child Health 1st Bl…  1,400 sqft
12  Gokul Lake View Apartments,  Gokul Lake View A…  1,852 sqft
13  Mayitri Enclave,5T Block, 4th T Block East, Ja…  1,350 sqft
14  B and B Opulent Spire,  Near Bharat Petroleum …  2,899 sqft
15  Independent House, 224, 7th Cross, 1st stage I…  5,000 sqft
16  4th Block  near Vivekananda Educational Centre…  2,114 sqft
17  Independent House, Tilak Nagar, near S.K. Bake…  2,500 sqft

                emi        price
0   1.43 Lacs/Month   2.5 Crores
1      65,911/Month  1.15 Crores
2      57,314/Month      1 Crore
3   1.71 Lacs/Month  2.99 Crores
4      83,106/Month  1.45 Crores
5   1.29 Lacs/Month  2.25 Crores
6      54,448/Month      95 Lacs
7   1.03 Lacs/Month   1.8 Crores
8   5.13 Lacs/Month  8.95 Crores
9   3.21 Lacs/Month   5.6 Crores
10     80,240/Month   1.4 Crores
11     68,777/Month   1.2 Crores
12     97,434/Month   1.7 Crores
13     48,717/Month      85 Lacs
14  2.98 Lacs/Month  5.19 Crores
15  5.44 Lacs/Month   9.5 Crores
16  1.32 Lacs/Month   2.3 Crores
17  1.43 Lacs/Month   2.5 Crores
```

2.3.6 to scrape first 10 product details which include product name , price , Image URL from https://www.bewakoof.com/bestseller?sort=popular .

[ ]:
```
def get_product_details(url):
    edge_options=Options()
    edge_options.add_argument("--headless")
    driver=webdriver.Edge(edge_options)
    driver.get(url)
    time.sleep(5)
    html=driver.page_source
    soup=BeautifulSoup(html, 'lxml')
    product_details=soup.find_all('div', class_='productCardBox')
    product_df=pd.DataFrame(columns=['Name', 'Price_in_INR', 'Image_URL'])
```

```
    for i, product_detail in enumerate(product_details):
        if i==10:
            break
        img_url=product_detail.find('div', class_='productImg').img['src']
        name=product_detail.find('div', class_='productNaming bkf-ellipsis').
 ↪text.strip()
        price=float(product_detail.find('div', class_='discountedPriceText␣
 ↪clr-p-black false').text.strip().replace(' ', ''))
        product_df.loc[i]=[name, price, img_url]
    return product_df.reset_index(drop=True)

url='https://www.bewakoof.com/bestseller?sort=popular'
get_product_details(url=url)
```

[ ]:

	Name	Price_in_INR	Image_URL
0	Bewakoof®Women's Pink & White Camo Printed Ove…	599.0	https://images.bewakoof.com/t640/women-s-pink-…
1	Bewakoof®Women's White Bored Typography Oversi…	549.0	https://images.bewakoof.com/t640/women-white-p…
2	bewakoof x disneyWomen's White & Black All Ove…	379.0	https://images.bewakoof.com/t640/women-s-white…
3	bewakoof x garfieldWomen's White All Over I Ha…	379.0	https://images.bewakoof.com/t640/women-aop-ove…
4	Bewakoof®Women's White Camo Printed Oversized …	494.0	https://images.bewakoof.com/t640/women-aop-ove…
5	bewakoof x looney tunesWomen's White All Over …	499.0	https://images.bewakoof.com/t640/women-aop-ove…
6	Bewakoof®Men's Blue Peace Not War Graphic Prin…	549.0	https://images.bewakoof.com/t640/men-s-blue-pe…
7	bewakoof x nasaMen's Black NASA Out Of The Spa…	599.0	https://images.bewakoof.com/t640/men-s-black-n…
8	bewakoof x dcMen's Black Batman Riddle (Bml) O…	629.0	https://images.bewakoof.com/t640/men-s-black-b…
9	Bewakoof®Men's Black Killmonger Graphic Printe…	519.0	https://images.bewakoof.com/t640/men-s-black-k…

2.3.7 Please visit https://www.cnbc.com/world/?region=world and scrap

- Headings
- Date
- News link

[ ]:
```
def get_news(url):
    html=req.get(url=url).content
    soup=BeautifulSoup(html, 'lxml')
```

```
    headings=soup.find('div', id='Home Page International-riverPlus')
    div_ids=set(re.
 ↪compile(pattern='HomePageInternational-riverPlus-11-\d{1,2}').
 ↪findall(headings.prettify()))
    news_df=pd.DataFrame(columns=['Heading', 'Date', 'News_link'])
    for i, div_id in enumerate(div_ids):
        news_details=headings.find('div', id=div_id)
        if news_details.find('div', class_='RiverHeadline-headline␣
 ↪RiverHeadline-hasThumbnail') is not None:
            heading=news_details.find('div', class_='RiverHeadline-headline␣
 ↪RiverHeadline-hasThumbnail').text.strip()
            link=news_details.find('div', class_='RiverHeadline-headline␣
 ↪RiverHeadline-hasThumbnail').a['href']
            date=news_details.find('span', class_='RiverByline-datePublished')
            if date is not None:
                date=date.text.strip()
                if 'ago' in date:
                    date=datetime.today().strftime('%d-%m-%Y')
            else:
                date=None
            news_df.loc[i]=[heading, date, link]
    return news_df.reset_index(drop=True)

url='https://www.cnbc.com/world/?region=world'
get_news(url=url)
```

[ ]:

	Heading	Date	\
0	'The frenzy continues': Researcher names 3 new…	29-03-2024	
1	Standard Chartered says Japan 'very, very clos…	None	
2	Japan and China stocks rise, while most Asia-P…	29-03-2024	
3	Baltimore disaster may be the largest-ever mar…	None	
4	The wealth of the 1% just hit a record $44 tri…	None	
5	Reddit shares plunge almost 25% in two days, f…	29-03-2024	
6	Xiaomi releases electric car $4K cheaper than …	None	
7	How this bull market could unravel and what to…	None	
8	Don't sweat the prospect of no Fed rate cuts, …	None	
9	Sam Bankman-Fried sentenced to 25 years for FT…	None	
10	eVTOLS: How flying cars are becoming reality	None	
11	China lifts tariffs on Australian wine, ends t…	None	
12	Toyota could introduce electric, plug-in Tacom…	None	
13	A new Japanese restaurant tops 'Asia's 50 Best…	None	
14	UBS chief's surprise return to the Swiss banki…	None	
15	Putin says NATO won't be attacked but F-16s wi…	None	
16	European stocks log gains of around 7% for the…	None	
17	Russia knew of Moscow terrorist attack plot we…	None	
18	China's Xi tells U.S. CEOs that bilateral rela…	None	
19	Italy's new 'Orient Express' isn't running yet…	29-03-2024	

```
                                                News_link
0                                                    /pro/
1    https://www.cnbc.com/2024/03/28/standard-chart…
2    https://www.cnbc.com/2024/03/29/asia-markets-l…
3    https://www.cnbc.com/2024/03/28/baltimore-disa…
4    https://www.cnbc.com/2024/03/28/wealth-of-the-…
5    https://www.cnbc.com/2024/03/28/reddit-shares-…
6    https://www.cnbc.com/2024/03/28/xiaomi-release…
7    https://www.cnbc.com/2024/03/28/how-this-bull-…
8    https://www.cnbc.com/2024/03/28/dont-sweat-the…
9    https://www.cnbc.com/video/2024/03/28/sam-bank…
10   https://www.cnbc.com/video/2024/03/28/evtols-h…
11   https://www.cnbc.com/2024/03/28/china-lifts-ta…
12   https://www.cnbc.com/2024/03/28/toyota-weighin…
13   https://www.cnbc.com/2024/03/28/what-are-the-b…
14   https://www.cnbc.com/2024/03/28/ubs-chiefs-sur…
15   https://www.cnbc.com/2024/03/28/ukraine-war-li…
16   https://www.cnbc.com/2024/03/28/european-marke…
17   https://www.cnbc.com/2024/03/28/russia-knew-of…
18   https://www.cnbc.com/2024/03/28/chinas-xi-tell…
19   https://www.cnbc.com/2024/03/29/la-dolce-vita-…
```

2.3.8 Please visit https://www.keaipublishing.com/en/journals/artificial-intelligence-in-agriculture/most-downloaded-articles/ and scrap-

- Paper title
- Date
- Author

[ ]:
```
def get_paper_details(url):
    html=req.get(url=url).content
    soup=BeautifulSoup(html, 'lxml')
    paper_details=soup.findAll('div', class_='article-listing')
    paper_df=pd.DataFrame(columns=['Title', 'Date', 'Author'])
    for i, paper_detail in enumerate(paper_details):
        title=paper_detail.find('h2', class_='h5 article-title').a.text.strip()
        date=paper_detail.find('p', class_='article-date').text.strip()
        author=paper_detail.find('p', class_='article-authors').text.strip()
        paper_df.loc[i]=[title, date, author]
    return paper_df.reset_index(drop=True)

url='https://www.keaipublishing.com/en/journals/
 ↪artificial-intelligence-in-agriculture/most-downloaded-articles/'
get_paper_details(url=url)
```

[ ]:

	Title	Date	Author
0	Implementation of artificial intelligence in a…	2020	Tanha Talaviya | Dhara Shah | Nivedita Patel…
1	A comprehensive review on automation in agricu…	June 2019	Kirtan Jha | Aalap Doshi | Poojan Patel | M…
2	Review of agricultural IoT technology	2022	Jinyuan Xu | Baoxing Gu | Guangzhao Tian
3	Automation and digitization of agriculture usi…	2021	A. Subeesh | C.R. Mehta
4	Real-time hyperspectral imaging for the in-fie…	2020	Zongmei Gao | Yuanyuan Shao | Guantao Xuan |…
5	Applications of electronic nose (e-nose) and e…	2020	Juzhong Tan | Jie Xu
6	Fruit ripeness classification: A survey	March 2023	Matteo Rizzo | Matteo Marcuzzo | Alessandro …
7	A review of imaging techniques for plant disea…	2020	Vijai Singh | Namita Sharma | Shikha Singh
8	Deep learning based computer vision approaches…	2022	V.G. Dhanya | A. Subeesh | N.L. Kushwaha | …
9	Comparison of CNN-based deep learning architec…	September 2023	Md Taimur Ahad | Yan Li | Bo Song | Touhid …
10	Transfer Learning for Multi-Crop Leaf Disease …	2022	Ananda S. Paymode | Vandana B. Malode
11	DeepRice: A deep learning and deep feature bas…	March 2024	P. Isaac Ritharson | Kumudha Raimond | X. An…
12	Plant disease detection using hybrid model bas…	2021	Punam Bedi | Pushkar Gole
13	How artificial intelligence uses to achieve th…	June 2023	Vilani Sachithra | L.D.C.S. Subhashini
14	Deep convolutional neural network models for w…	2022	A. Subeesh | S. Bhole | K. Singh | N.S. Cha…
15	Machine learning in nutrient management: A review	September 2023	Oumnia Ennaji | Leonardus Vergütz | Achraf E…
16	Machine learning for weed-plant discrimination…	December 2023	Filbert H. Juwono | W.K. Wong | Seema Verma …
17	A systematic review of machine learning techni…	2022	Md Ekramul Hossain | Muhammad Ashad Kabir | …
18	Examining the interplay between artificial int…	2022	Abderahman Rejeb | Karim Rejeb | Suhaiza Zai…
19	A review on computer vision systems in monitor…	2020	
20	Artificial cognition for applications in smart…	2020	
21	Explainable artificial intelligence and interp…	2022	
22	Vision Intelligence for Smart Sheep Farming: A…	March 2024	
23	Crop diagnostic system: A robust disease detec…	December 2023	
24	Blockchain: A new safeguard for agri-foods	2020	

```
19  Cedric Okinda |  Innocent Nyalala |  Tchalla K…
20  Misbah Pathan |  Nivedita Patel |  Hiteshri Ya…
21                                      Masahiro Ryo
22  Galib Muhammad Shahriar Himel |  Md. Masudul I…
23               R. Abbasi |  P. Martinez |  R. Ahmad
24    Jie Xu |  Shuang Guo |  David Xie |  Yaxuan Yan
```