• Read all the problem statements, notes carefully and scrape the required data using any web scraping tool of your choice.

• You have to handle commonly occurring EXCEPTIONS by using exception handling programing. To get information about selenium Exceptions. You may visit following links:

#1. https://selenium-python.readthedocs.io/api.html

2. https://www.guru99.com/exception-handling-selenium.html

3. https://stackoverflow.com/questions/38022658/selenium-python-handling-no-such-elementexception/38023345

1. Scrape the details of most viewed videos on YouTube from Wikipedia. Url = https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos You need to find following details:

A) Rank

B) Name

C) Artist

D) Upload date

E) Views

```
!pip install bs4
!pip install requests
import warnings
warnings.filterwarnings('ignore')
```

```
Requirement already satisfied: bs4 in c:\users\anand\anaconda3\lib\
site-packages (0.0.2)
Requirement already satisfied: beautifulsoup4 in c:\users\anand\
anaconda3\lib\site-packages (from bs4) (4.12.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\anand\
anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.4)
Requirement already satisfied: requests in c:\users\anand\anaconda3\
lib\site-packages (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\
anand\anaconda3\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\anand\
anaconda3\lib\site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\anand\
anaconda3\lib\site-packages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\anand\
anaconda3\lib\site-packages (from requests) (2024.2.2)
```

```
from selenium import webdriver
import time
from selenium.common.exceptions import NoSuchElementException #
importing exception
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By

# importing the required libraries
from bs4 import BeautifulSoup
import requests

url = 'https://en.wikipedia.org/wiki/List_of_most-
viewed_YouTube_videos'

response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

table = soup.find('table', {'class':'wikitable'})
```

```
for row in table.find_all('tr')[1:20]:
    columns = row.find_all('td')
    try:

        rank = columns[0].text.strip()
        name = columns[1].text.strip()
        artist = columns[1].text.strip()
        upload_date = columns[3].text.strip()
        views = columns[2].text.strip()

        print(f"Rank: {rank}")
        print(f"Name: {name}")
        print(f"Artist: {artist}")
        print(f"Upload Date: {upload_date}")
        print(f"Views: {views}")
        print("")
    except NoSuchElementException:
        Expected_Delivery.append('-')
```

```
Rank: "Baby Shark Dance"[6]
Name: Pinkfong Baby Shark - Kids' Songs & Stories
Artist: Pinkfong Baby Shark - Kids' Songs & Stories
Upload Date: June 17, 2016
Views: 14.09

Rank: "Despacito"[9]
Name: Luis Fonsi
Artist: Luis Fonsi
Upload Date: January 12, 2017
Views: 8.38

Rank: "Johny Johny Yes Papa"[17]
Name: LooLoo Kids - Nursery Rhymes and Children's Songs
Artist: LooLoo Kids - Nursery Rhymes and Children's Songs
Upload Date: October 8, 2016
Views: 6.87

Rank: "Bath Song"[18]
Name: Cocomelon - Nursery Rhymes
Artist: Cocomelon - Nursery Rhymes
Upload Date: May 2, 2018
Views: 6.62

Rank: "Shape of You"[19]
Name: Ed Sheeran
Artist: Ed Sheeran
Upload Date: January 30, 2017
Views: 6.20
```

Rank: "See You Again"[22]
Name: Wiz Khalifa
Artist: Wiz Khalifa
Upload Date: April 6, 2015
Views: 6.17

Rank: "Wheels on the Bus"[27]
Name: Cocomelon - Nursery Rhymes
Artist: Cocomelon - Nursery Rhymes
Upload Date: May 24, 2018
Views: 5.88

Rank: "Phonics Song with Two Words"[28]
Name: ChuChu TV Nursery Rhymes & Kids Songs
Artist: ChuChu TV Nursery Rhymes & Kids Songs
Upload Date: March 6, 2014
Views: 5.70

Rank: "Uptown Funk"[29]
Name: Mark Ronson
Artist: Mark Ronson
Upload Date: November 19, 2014
Views: 5.15

Rank: "Learning Colors – Colorful Eggs on a Farm"[30]
Name: Miroshka TV
Artist: Miroshka TV
Upload Date: February 27, 2018
Views: 5.07

Rank: "Gangnam Style"[31]
Name: Psy
Artist: Psy
Upload Date: July 15, 2012
Views: 5.05

Rank: "Masha and the Bear – Recipe for Disaster"[36]
Name: Get Movies
Artist: Get Movies
Upload Date: January 31, 2012
Views: 4.58

Rank: "Dame Tu Cosita"[37]
Name: Ultra Records
Artist: Ultra Records
Upload Date: April 5, 2018
Views: 4.55

Rank: "Axel F"[38]

Name: Crazy Frog
Artist: Crazy Frog
Upload Date: June 16, 2009
Views: 4.34

Rank: "Sugar"[39]
Name: Maroon 5
Artist: Maroon 5
Upload Date: January 14, 2015
Views: 4.00

Rank: "Counting Stars"[40]
Name: OneRepublic
Artist: OneRepublic
Upload Date: May 31, 2013
Views: 3.97

Rank: "Baa Baa Black Sheep"[41]
Name: Cocomelon - Nursery Rhymes
Artist: Cocomelon - Nursery Rhymes
Upload Date: June 25, 2018
Views: 3.96

Rank: "Roar"[42]
Name: Katy Perry
Artist: Katy Perry
Upload Date: September 5, 2013
Views: 3.96

Rank: "Lakdi Ki Kathi"[43]
Name: Jingle Toons
Artist: Jingle Toons
Upload Date: June 14, 2018
Views: 3.91

2. Scrape the details team India’s international fixtures from bcci.tv.

Url = https://www.bcci.tv/.

You need to find following details:

A) Series

B) Place

C) Date

D) Time

Note: - From bcci.tv home page you have reach to the international fixture page through code.

```
!pip install selenium
```

```
Collecting selenium
  Obtaining dependency information for selenium from
https://files.pythonhosted.org/packages/3f/fd/c2e7bb547b5b96c7bd536b4a
80c4564b7ce5cd38d10095fbba8648996ab9/selenium-4.18.1-py3-none-
any.whl.metadata
  Using cached selenium-4.18.1-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: urllib3[socks]<3,>=1.26 in c:\users\
anand\anaconda3\lib\site-packages (from selenium) (1.26.16)
Collecting trio~=0.17 (from selenium)
  Obtaining dependency information for trio~=0.17 from
https://files.pythonhosted.org/packages/14/fb/9299cf74953f473a15accfdb
e2c15218e766bae8c796f2567c83bae03e98/trio-0.24.0-py3-none-
any.whl.metadata
  Using cached trio-0.24.0-py3-none-any.whl.metadata (4.9 kB)
```

```
Collecting trio-websocket~=0.9 (from selenium)
  Obtaining dependency information for trio-websocket~=0.9 from
https://files.pythonhosted.org/packages/48/be/a9ae5f50cad5b6f85bd2574c
2c923730098530096e170c1ce7452394d7aa/trio_websocket-0.11.1-py3-none-
any.whl.metadata
  Using cached trio_websocket-0.11.1-py3-none-any.whl.metadata (4.7
kB)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\anand\
anaconda3\lib\site-packages (from selenium) (2023.7.22)
Collecting typing_extensions>=4.9.0 (from selenium)
  Obtaining dependency information for typing_extensions>=4.9.0 from
https://files.pythonhosted.org/packages/f9/de/dc04a3ea60b22624b51c703a
84bbe0184abcd1d0b9bc8074b5d6b7ab90bb/typing_extensions-4.10.0-py3-
none-any.whl.metadata
  Downloading typing_extensions-4.10.0-py3-none-any.whl.metadata (3.0
kB)
Requirement already satisfied: attrs>=20.1.0 in c:\users\anand\
anaconda3\lib\site-packages (from trio~=0.17->selenium) (22.1.0)
Requirement already satisfied: sortedcontainers in c:\users\anand\
anaconda3\lib\site-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in c:\users\anand\anaconda3\lib\
site-packages (from trio~=0.17->selenium) (3.4)
Collecting outcome (from trio~=0.17->selenium)
  Obtaining dependency information for outcome from
https://files.pythonhosted.org/packages/55/8b/5ab7257531a5d830fc8000c4
76e63c935488d74609b50f9384a643ec0a62/outcome-1.3.0.post0-py2.py3-none-
any.whl.metadata
  Using cached outcome-1.3.0.post0-py2.py3-none-any.whl.metadata (2.6
kB)
Collecting sniffio>=1.3.0 (from trio~=0.17->selenium)
  Obtaining dependency information for sniffio>=1.3.0 from
https://files.pythonhosted.org/packages/e9/44/75a9c9421471a6c4805dbf23
56f7c181a29c1879239abab1ea2cc8f38b40/sniffio-1.3.1-py3-none-
any.whl.metadata
  Using cached sniffio-1.3.1-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: cffi>=1.14 in c:\users\anand\anaconda3\
lib\site-packages (from trio~=0.17->selenium) (1.15.1)
Collecting wsproto>=0.14 (from trio-websocket~=0.9->selenium)
  Obtaining dependency information for wsproto>=0.14 from
https://files.pythonhosted.org/packages/78/58/e860788190eba3bcce367f74
d29c4675466ce8dddfba85f7827588416f01/wsproto-1.2.0-py3-none-
any.whl.metadata
  Using cached wsproto-1.2.0-py3-none-any.whl.metadata (5.6 kB)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\
users\anand\anaconda3\lib\site-packages (from urllib3[socks]<3,>=1.26-
>selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\anand\anaconda3\
lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Collecting h11<1,>=0.9.0 (from wsproto>=0.14->trio-websocket~=0.9-
```

```
>selenium)
  Obtaining dependency information for h11<1,>=0.9.0 from
https://files.pythonhosted.org/packages/95/04/ff642e65ad6b90db43e668d7
0ffb6736436c7ce41fcc549f4e9472234127/h11-0.14.0-py3-none-
any.whl.metadata
  Using cached h11-0.14.0-py3-none-any.whl.metadata (8.2 kB)
Using cached selenium-4.18.1-py3-none-any.whl (10.0 MB)
Using cached trio-0.24.0-py3-none-any.whl (460 kB)
Using cached trio_websocket-0.11.1-py3-none-any.whl (17 kB)
Downloading typing_extensions-4.10.0-py3-none-any.whl (33 kB)
Using cached sniffio-1.3.1-py3-none-any.whl (10 kB)
Using cached wsproto-1.2.0-py3-none-any.whl (24 kB)
Using cached outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)
Using cached h11-0.14.0-py3-none-any.whl (58 kB)
Installing collected packages: typing_extensions, sniffio, outcome,
h11, wsproto, trio, trio-websocket, selenium
  Attempting uninstall: typing_extensions
    Found existing installation: typing_extensions 4.7.1
    Uninstalling typing_extensions-4.7.1:
      Successfully uninstalled typing_extensions-4.7.1
  Attempting uninstall: sniffio
    Found existing installation: sniffio 1.2.0
    Uninstalling sniffio-1.2.0:
      Successfully uninstalled sniffio-1.2.0
Successfully installed h11-0.14.0 outcome-1.3.0.post0 selenium-4.18.1
sniffio-1.3.1 trio-0.24.0 trio-websocket-0.11.1 typing_extensions-
4.10.0 wsproto-1.2.0
```

```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome()


driver.get("https://www.bcci.tv/fixtures?
platform=international&type=men")

Series=[]
Place=[]
Date=[]
Time=[]

series_tags=driver.find_elements(By.XPATH,'//div[@class="match-time
no-margin ng-binding"]')
for i in series_tags:
```

```
    series=i.text
    Series.append(series)

place_tags=driver.find_elements(By.XPATH,'//span[@class="ng-binding
ng-scope"]')
for i in place_tags:
    place=i.text
    Place.append(place)
date_tags=driver.find_elements(By.XPATH,'//div[@class="match-dates ng-
binding"]')
for i in date_tags:
    date=i.text
    Date.append(date)
time_tags=driver.find_elements(By.XPATH,'//div[@class="match-time no-
margin ng-binding"]')
for i in time_tags:
    time=i.text
    Time.append(time)

print(len(Series),len(Place),len(Date),len(Time))
```

```
6 6 6 6
```

```
import pandas as pd
df=pd.DataFrame({'series':Series,'place':Place,'date':Date,'time':Time
})
df
```

```
        series                                                place
date  \
0  9:30 AM IST  Himachal Pradesh Cricket Association Stadium,  7
MARCH, 2024
1  8:00 PM IST                              Harare Sports Club,   6
JULY, 2024
2  8:00 PM IST                              Harare Sports Club,   7
JULY, 2024
3  8:00 PM IST                              Harare Sports Club,  10
JULY, 2024
4  8:00 PM IST                              Harare Sports Club,  13
JULY, 2024
5  8:00 PM IST                              Harare Sports Club,  14
JULY, 2024

          time
0  9:30 AM IST
1  8:00 PM IST
2  8:00 PM IST
3  8:00 PM IST
4  8:00 PM IST
5  8:00 PM IST
```

3. Scrape the details of State-wise GDP of India from statisticstime.com.

Url = http://statisticstimes.com/

You have to find following details: A) Rank

B) State

C) GSDP(18-19)- at current prices

D) GSDP(19-20)- at current prices

E) Share(18-19)

F) GDP($ billion)

Note: - From statisticstimes home page you have to reach to economy page through code.

```
!pip install selenium

Requirement already satisfied: selenium in c:\users\anand\anaconda3\
lib\site-packages (4.18.1)
Requirement already satisfied: urllib3[socks]<3,>=1.26 in c:\users\
anand\anaconda3\lib\site-packages (from selenium) (1.26.16)
Requirement already satisfied: trio~=0.17 in c:\users\anand\anaconda3\
lib\site-packages (from selenium) (0.24.0)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\anand\
anaconda3\lib\site-packages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\anand\
anaconda3\lib\site-packages (from selenium) (2023.7.22)
```

```
Requirement already satisfied: typing_extensions>=4.9.0 in c:\users\
anand\anaconda3\lib\site-packages (from selenium) (4.10.0)
Requirement already satisfied: attrs>=20.1.0 in c:\users\anand\
anaconda3\lib\site-packages (from trio~=0.17->selenium) (22.1.0)
Requirement already satisfied: sortedcontainers in c:\users\anand\
anaconda3\lib\site-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in c:\users\anand\anaconda3\lib\
site-packages (from trio~=0.17->selenium) (3.4)
Requirement already satisfied: outcome in c:\users\anand\anaconda3\
lib\site-packages (from trio~=0.17->selenium) (1.3.0.post0)
Requirement already satisfied: sniffio>=1.3.0 in c:\users\anand\
anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.3.1)
Requirement already satisfied: cffi>=1.14 in c:\users\anand\anaconda3\
lib\site-packages (from trio~=0.17->selenium) (1.15.1)
Requirement already satisfied: wsproto>=0.14 in c:\users\anand\
anaconda3\lib\site-packages (from trio-websocket~=0.9->selenium)
(1.2.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\
users\anand\anaconda3\lib\site-packages (from urllib3[socks]<3,>=1.26-
>selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\anand\anaconda3\
lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\anand\
anaconda3\lib\site-packages (from wsproto>=0.14->trio-websocket~=0.9-
>selenium) (0.14.0)
```

```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome()


driver.get("https://statisticstimes.com/economy/india/indian-states-
gdp.php")

Rank=[]
State=[]
GSDP_18_19=[]
GSDP_19_20=[]
Share_18_19=[]
gdp_billion=[]

rank_tags=driver.find_elements(By.XPATH,'//tr[@class="odd"]//td[1]')
for i in rank_tags:
    rank=i.text
```

```
    Rank.append(rank)
state_tags=driver.find_elements(By.XPATH,'//tr[@class="odd"]//td[2]')
for i in state_tags:
    state=i.text
    State.append(state)

gsdp_tags=driver.find_elements(By.XPATH,'//tr[@class="odd"]//td[3]')
for i in gsdp_tags:
    gsdp=i.text
    GSDP_18_19.append(gsdp)
gsdps_tags=driver.find_elements(By.XPATH,'//tr[@class="odd"]//td[4]')
for i in gsdps_tags:
    gsdps=i.text
    GSDP_19_20.append(gsdps)
share_tags=driver.find_elements(By.XPATH,'//tr[@class="odd"]//td[5]')
for i in share_tags:
    share=i.text
    Share_18_19.append(share)
gdp_tags=driver.find_elements(By.XPATH,'//tr[@class="odd"]//td[6]')
for i in gdp_tags:
    gdp=i.text
    gdp_billion.append(gdp)

print(len(Rank),len(State),len(GSDP_18_19),len(GSDP_19_20),len(Share_1
8_19),len(gdp_billion))
```

```
34 34 34 34 34 34
```

```
import pandas as pd
df=pd.DataFrame({'rank':Rank,'state':State,'Gsdp_18_!
9':GSDP_18_19,'Gsdp_19_20':GSDP_19_20,'share':Share_18_19,'Gdp':gdp_bi
llion})
df
```

```
   rank                        state Gsdp_18_!9 Gsdp_19_20   share
Gdp
0     1                  Maharashtra          -  3,108,022  13.24%
417.163
1     3                Uttar Pradesh  2,257,575  1,974,532   8.41%
265.024
2     5                      Gujarat          -  1,937,066   8.25%
259.996
3     7                    Rajasthan  1,413,620  1,218,193   5.19%
163.507
4     9                Andhra Pradesh  1,317,728  1,133,837   4.83%
152.185
5    11                       Kerala          -    932,470   3.97%
125.157
6    13                      Haryana    994,154    870,665   3.71%
116.862
```

```
7    15                      Bihar    751,396    650,302   2.77%
87.284
8    17                      Assam    493,167    412,612   1.76%
55.381
9    19                  Jharkhand    393,722    358,863   1.53%
48.167
10   21          Jammu & Kashmir-UT    227,927    199,917   0.85%
26.833
11   23                        Goa          -     82,604   0.35%
11.087
12   25                  Chandigarh          -     45,635   0.19%
6.125
13   27                   Meghalaya     42,697     38,785   0.17%
5.206
14   29                     Manipur          -     36,594   0.16%
4.912
15   31                    Nagaland          -     31,913   0.14%
4.283
16   33  Andaman & Nicobar Islands          -     10,371   0.04%
1.392
17    1                 Maharashtra          -  2,690,525  13.24%
-
18    3                   Karnataka  2,036,748  1,782,121   8.77%
1,190,851
19    5                     Gujarat          -  1,700,504   8.37%
-
20    7                   Rajasthan  1,259,527  1,084,845   5.34%
694,771
21    9                   Telangana  1,187,082  1,024,205   5.04%
642,207
22   11                      Kerala          -    822,970   4.05%
-
23   13                     Haryana    885,308    779,197   3.83%
542,972
24   15                      Odisha    691,888    587,900   2.89%
403,222
25   17                      Assam    427,147    363,161   1.79%
251,688
26   19                  Jharkhand    360,689    325,830   1.60%
235,685
27   21          Jammu & Kashmir-UT    185,489    162,644   0.80%
105,636
28   23                        Goa          -     73,973   0.36%
-
29   25                  Chandigarh          -     40,573   0.20%
-
30   27                   Meghalaya     37,925     34,441   0.17%
21,905
31   29           Arunachal Pradesh          -     31,669   0.16%
```

```
-
32   31                    Nagaland          -     27,859   0.14%
-
33   33  Andaman & Nicobar Islands          -      9,209   0.05%
-
```

4. Scrape the details of trending repositories on Github.com.

Url = https://github.com/

You have to find the following details:

A) Repository title

B) Repository description

C) Contributors count

D) Language used

```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome()

driver.get('https://github.com/trending')

# scrape all product urls
product_urls=[]
start=0
```

```
end=1
for page in range(start,end):
    urls=driver.find_elements(By.XPATH,'//a[@class="Link"]')
    for i in urls:
        product_urls.append(i.get_attribute('href'))
    time.sleep(2)

print(len(product_urls))
```

```
25
```

```
import pandas as pd
df=pd.DataFrame({'product_urls':product_urls})
df
```

	product_urls
0	https://github.com/cloudflare/pingora
1	https://github.com/Lissy93/web-check
2	https://github.com/Lunakepio/Mario-Kart-3.js
3	https://github.com/dockur/windows
4	https://github.com/yuzu-emu/yuzu
5	https://github.com/pmndrs/uikit
6	https://github.com/HumanAIGC/EMO
7	https://github.com/moom825/xeno-rat
8	https://github.com/kyegomez/BitNet
9	https://github.com/microsoft/Security-101
10	https://github.com/HumanAIGC/AnimateAnyone
11	https://github.com/Pawdroid/Free-servers
12	https://github.com/aishwaryanr/awesome-generat...
13	https://github.com/wazuh/wazuh
14	https://github.com/cloudcommunity/Free-Certifi...
15	https://github.com/redis/ioredis
16	https://github.com/evo-design/evo
17	https://github.com/pure-admin/vue-pure-admin
18	https://github.com/jwasham/coding-interview-un...
19	https://github.com/myshell-ai/MeloTTS
20	https://github.com/yuzu-emu/yuzu-android
21	https://github.com/bruin-data/ingestr
22	https://github.com/ossu/computer-science
23	https://github.com/polyfillpolyfill/polyfill-s...
24	https://github.com/DataTalksClub/data-engineer...

```
Repository_title=[]
Repository_description=[]
Contributors_count=[]
Language_used=[]

repository_tags=driver.find_elements(By.XPATH,'//span[@class="text-
normal"]')
for i in repository_tags:
    titles=i.text
```

```
    Repository_title.append(titles)

repository_description_tags=driver.find_elements(By.XPATH,'//p[@class="col-9 color-fg-muted my-1 pr-4"]')
for i in repository_description_tags:
    description=i.text
    Repository_description.append(description)

contribution_tags=driver.find_elements(By.XPATH,'//span[@class="d-inline-block float-sm-right"]')
for i in contribution_tags:
    contribution=i.text
    Contributors_count.append(contribution)

language_tags=driver.find_elements(By.XPATH,'//span[@itemprop="programmingLanguage"]')
for i in language_tags:
    language=i.text
    Language_used.append(language_tags)

print(len(Repository_title),len(Repository_description),len(Contributors_count),len(Language_used),len(product_urls))
```

```
24 24 24 24 25
```

```
import pandas as pd
df=pd.DataFrame({'repository_title':Repository_title,'repository_description':Repository_description,'contribution_count':Contributors_count,'language':Language_used})
df
```

```
       repository_title
repository_description  \
0            vvbbnn00 /  该项目可以让你通过订阅的方式使用Cloudflare WARP+，自动获取流量。This p...
1               google /  The official PyTorch implementation of Google'...
2             yuzu-emu /                              Nintendo Switch emulator
3           SoraWebui /  SoraWebui is an open-source Sora web client, e...
4               google /  lightweight, standalone C++ inference engine f...
5              levihsu /  Official implementation of OOTDiffusion: Outfi...
6     google-deepmind /              Open weights LLM from Google DeepMind.
7       gunnarmorling /  ▢▢🐝🦀The One Billion Row Challenge -- A fun ...
8           massgravel /  A Windows and Office activator using HWID /
```

```
Oh...
9               qarmin /  Multi functional app to find duplicates,
empty...
10           facebook /  React Strict DOM (RSD) is a subset of React
DO...
11          jackfrued /                                  Python - 100天
从新手到大师
12            lobehub /  🤯 Lobe Chat - an open-source, modern-design
Ch...
13          movie-web /  A small web app for watching movies and shows
...
14           pydantic /                             Build better UIs
faster.
15        public-apis /                      A collective list of free
APIs
16             mut-ex /  An intuitive GUI for GLIGEN that uses ComfyUI
...
17         chenzomi12 /  Deep Learning System core principles
introduct...
18           MHSanaei /  Xray panel supporting multi-protocol multi-
use...
19           taikoxyz /                                    A based
rollup. 🥁
20      FujiwaraChoki /       Automate the process of making money
online.
21          microsoft /  18 Lessons, Get Started Building with
Generati...
22             ollama /  Get up and running with Llama 2, Mistral,
Gemm...
23  openai-translator /  基于 ChatGPT API 的划词翻译浏览器插件和跨平台桌面端应
用 - Browser e...

       contribution_count
language
0   3,244 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
1   2,433 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
2   4,576 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
3   1,117 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
4   3,686 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
5     993 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
6     728 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
7     423 stars this week
```

```
[<selenium.webdriver.remote.webelement.WebElem...
8   2,839 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
9   1,021 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
10    797 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
11    973 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
12  1,169 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
13    698 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
14    419 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
15  3,700 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
16    715 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
17    310 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
18    630 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
19    652 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
20    553 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
21    999 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
22  3,702 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
23    739 stars this week
[<selenium.webdriver.remote.webelement.WebElem...
```

5. Scrape the details of top 100 songs on billiboard.com. Url = https:/www.billboard.com/
You have to find the following details:

A) Song name

B) Artist name

C) Last week rank

D) Peak rank

E) Weeks on board

Note: - From the home page you have to click on the charts option then hot 100-page link through code.

```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome()

driver.get('https://www.billboard.com/')

search=driver.find_element(By.XPATH,"/html/body/div[3]/header/div/
div[2]/div/div/div[2]/div[2]/div/div/nav/ul/li[1]/a")
search.click()
```

```
Song_name=[]
Artist_name=[]
Last_week_rank=[]
Peak_rank=[]
Weeks_on_board=[]

song_tags=driver.find_elements(By.XPATH,'//h3[@class="c-title  a-no-
trucate a-font-primary-bold-s u-letter-spacing-0021 lrv-u-font-size-
18@tablet lrv-u-font-size-16 u-line-height-125 u-line-height-
normal@mobile-max a-truncate-ellipsis u-max-width-330 u-max-width-
230@tablet-only"]')
for i in song_tags:
    song=i.text
    Song_name.append(song)

artist_tags=driver.find_elements(By.XPATH,'//span[@class="c-label  a-
no-trucate a-font-primary-s lrv-u-font-size-14@mobile-max u-line-
height-normal@mobile-max u-letter-spacing-0021 lrv-u-display-block a-
truncate-ellipsis-2line u-max-width-330 u-max-width-230@tablet-
only"]')
for i in artist_tags:
    artist=i.text
    Artist_name.append(artist)

last_week_tags=driver.find_elements(By.XPATH,'//li[@class="o-chart-
results-list__item // a-chart-color u-width-72 u-width-55@mobile-max
u-width-55@tablet-only lrv-u-flex lrv-u-flex-shrink-0 lrv-u-align-
items-center lrv-u-justify-content-center lrv-u-border-b-1 u-border-b-
0@mobile-max lrv-u-border-color-grey-light u-background-color-white-
064@mobile-max u-hidden@mobile-max"][1]')
for i in last_week_tags:
    last_week=i.text
    Last_week_rank.append(last_week)

peak_rank_tags=driver.find_elements(By.XPATH,'//li[@class="o-chart-
results-list__item // a-chart-bg-color a-chart-color u-width-72 u-
width-55@mobile-max u-width-55@tablet-only lrv-u-flex lrv-u-flex-
shrink-0 lrv-u-align-items-center lrv-u-justify-content-center lrv-u-
background-color-grey-lightest lrv-u-border-b-1 u-border-b-0@mobile-
max lrv-u-border-color-grey-light u-hidden@mobile-max"][2]')
for i in peak_rank_tags:
    peak_rank=i.text
    Peak_rank.append(peak_rank)

week_tags=driver.find_elements(By.XPATH,'//li[@class="o-chart-results-
list__item // a-chart-color u-width-72 u-width-55@mobile-max u-width-
55@tablet-only lrv-u-flex lrv-u-flex-shrink-0 lrv-u-align-items-center
lrv-u-justify-content-center lrv-u-border-b-1 u-border-b-0@mobile-max
lrv-u-border-color-grey-light u-background-color-white-064@mobile-max
u-hidden@mobile-max"][2]')
```

```
for i in week_tags:
    week=i.text
    Weeks_on_board.append(week)

print(len(Song_name),len(Artist_name),len(Last_week_rank),len(Peak_ran
k),len(Weeks_on_board))
```

```
199 199 199 199 199
```

```
import pandas as pd
df=pd.DataFrame({'song_name':Song_name,'artist_name':Artist_name,'last
_weel_rank':Last_week_rank,'peak_rank':Peak_rank,'week_on_borad':Weeks
_on_board})
df
```

```
                     song_name  \
0             Beautiful Things
1                     Carnival
2                 Lose Control
3                 Cruel Summer
4               Training Season
..                          ...
194  Where The Wild Things Are
195            Ella Baila Sola
196                   Nonsense
197              See You Again
198               La Intencion

                                          artist_name last_weel_rank
\
0                                       Benson Boone               1

1    ¥$: Kanye West & Ty Dolla $ign Featuring Rich ...               2

2                                        Teddy Swims               7

3                                       Taylor Swift               3

4                                           Dua Lipa               -

..                                                ...             ...

194                                       Luke Combs               -

195                          Eslabon Armado X Peso Pluma             189

196                                  Sabrina Carpenter               -

197            Tyler, The Creator Featuring Kali Uchis             194

198                         Christian Nodal & Peso Pluma             175
```

```
    peak_rank week_on_borad
0           1             5
1           2             2
2           4            23
3           1            49
4           6             1
..        ...           ...
194       196             1
195         1            49
196        36            37
197        35            44
198        96             4

[199 rows x 5 columns]
```

6. Scrape the details of Highest selling novels.

A) Book name

B) Author name

C) Volumes sold

D) Publisher

E) Genre

Url - https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-grey-compare

```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome()

driver.get('
https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-
books-all-time-fifty-shades-grey-compare')

Book_name=[]
Author_name=[]
Volumes_sold=[]
```

```
Publisher=[]
Genre=[]

book_name_tags=driver.find_elements(By.XPATH,'//table[@class="in-
article sortable"]//tbody//tr//td[2]')
for i in book_name_tags:
    book=i.text
    Book_name.append(book)

author_name_tags=driver.find_elements(By.XPATH,'//table[@class="in-
article sortable"]//tbody//tr//td[3]')
for i in author_name_tags:
    author=i.text
    Author_name.append(author)

volumn_sold_tags=driver.find_elements(By.XPATH,'//table[@class="in-
article sortable"]//tbody//tr//td[4]')
for i in volumn_sold_tags:
    volumn=i.text
    Volumes_sold.append(volumn)

publisher_tags=driver.find_elements(By.XPATH,'//table[@class="in-
article sortable"]//tbody//tr//td[5]')
for i in publisher_tags:
    publisher=i.text
    Publisher.append(publisher)

genre_tags=driver.find_elements(By.XPATH,'//table[@class="in-article
sortable"]//tbody//tr//td[6]')
for i in genre_tags:
    genre=i.text
    Genre.append(genre)


print(len(Book_name),len(Author_name),len(Volumes_sold),len(Publisher)
,len(Genre))
```

```
100 100 100 100 100
```

```
import pandas as pd
df=pd.DataFrame({'book_name':Book_name,'author_name':Author_name,'volu
mn_sold':Volumes_sold,'publisher':Publisher,'genre':Genre})
df
```

```
                                              book_name
author_name  \
0                                  Da Vinci Code,The        Brown,
Dan
1               Harry Potter and the Deathly Hallows     Rowling,
J.K.
2            Harry Potter and the Philosopher's Stone     Rowling,
```

```
J.K.
3           Harry Potter and the Order of the Phoenix     Rowling,
J.K.
4                                Fifty Shades of Grey      James, E.
L.
..                                                  ...                ..
.
95                                          Ghost,The    Harris,
Robert
96                      Happy Days with the Naked Chef     Oliver,
Jamie
97              Hunger Games,The:Hunger Games Trilogy  Collins,
Suzanne
98  Lost Boy,The:A Foster Child's Search for the L...      Pelzer,
Dave
99  Jamie's Ministry of Food:Anyone Can Learn to C...     Oliver,
Jamie

   volumn_sold        publisher                         genre
0    5,094,805       Transworld  Crime, Thriller & Adventure
1    4,475,152       Bloomsbury           Children's Fiction
2    4,200,654       Bloomsbury           Children's Fiction
3    4,179,479       Bloomsbury           Children's Fiction
4    3,758,936     Random House              Romance & Sagas
..         ...              ...                         ...
95     807,311     Random House   General & Literary Fiction
96     794,201          Penguin        Food & Drink: General
97     792,187  Scholastic Ltd.          Young Adult Fiction
98     791,507            Orion           Biography: General
99     791,095          Penguin        Food & Drink: General

[100 rows x 5 columns]
```

7. Scrape the details most watched tv series of all time from imdb.com.

Url = https://www.imdb.com/list/ls095964455/
You have to find the following details:

A) Name

B) Year span

C) Genre

D) Run time

E) Ratings

F) Votes

```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome()

driver.get('https://www.imdb.com/list/ls512407256/')

NAME=[]
GENRE=[]
RUNTIME=[]
RATING=[]
VOTES=[]

name_tags=driver.find_elements(By.XPATH,'//h3[@class="lister-item-
header"]//a')
for i in name_tags:
    name=i.text
    NAME.append(name)

genre_tags=driver.find_elements(By.XPATH,'//span[@class="genre"]')
for i in genre_tags:
    genre=i.text
    GENRE.append(genre)

runtime_tags=driver.find_elements(By.XPATH,'//span[@class="runtime"]')
for i in runtime_tags:
    runtime=i.text
    RUNTIME.append(runtime)

rating_tags=driver.find_elements(By.XPATH,'//div[@class="ipl-rating-
star small"]//span[2]')
for i in rating_tags:
    rating=i.text
    RATING.append(rating)



voting_tags=driver.find_elements(By.XPATH,'//div[@class="lister-item-
content"]//p[4]//span[2]')
for i in voting_tags:
```

```
    voting=i.text
    VOTES.append(voting)



print(len(NAME),len(GENRE),len(RUNTIME),len(RATING),len(VOTES))
```

```
100 100 100 100 100
```

```
import pandas as pd
df=pd.DataFrame({'name':NAME,'genre':GENRE,'runtime':RUNTIME,'rating':
RATING,'votes':VOTES})
df
```

```
                    name                       genre runtime rating
votes
0        Game of Thrones  Action, Adventure, Drama  55 min    9.2
2,262,355
1        Stranger Things    Drama, Fantasy, Horror  51 min    8.7
1,320,331
2       The Walking Dead   Drama, Horror, Thriller  44 min    8.1
1,072,174
3         13 Reasons Why  Drama, Mystery, Thriller  60 min    7.5
313,460
4                The 100    Drama, Mystery, Sci-Fi  43 min    7.6
273,387
..                   ...                       ...     ...    ...
...
95        True Detective     Crime, Drama, Mystery  55 min    8.9
645,444
96             Teen Wolf    Action, Drama, Fantasy  41 min    7.7
162,050
97                The OA   Drama, Fantasy, Mystery  60 min    7.8
114,868
98          The Simpsons         Animation, Comedy  22 min    8.7
433,070
99  Desperate Housewives    Comedy, Drama, Mystery  45 min    7.6
138,683

[100 rows x 5 columns]
```

8. Details of Datasets from UCI machine learning repositories.

Url = https://archive.ics.uci.edu/ You have to find the following details:

A) Dataset name

B) Data type

C) Task

D) Attribute type

E) No of instances

F) No of attribute G) Year

Note: - from the home page you have to go to the Show All Dataset page through code.

```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome()
```

```
driver.get('https://archive.ics.uci.edu/')


product_urls=[]
start=0
end=5
for page in range(start,end):
    url=driver.find_elements(By.XPATH,'//a[@class="link-hover link
text-xl font-semibold"]')
    for i in url:
        product_urls.append(i.get_attribute('href'))
    time.sleep(2)

print(len(product_urls))
```

```
60
```

```
import pandas as pd
df=pd.DataFrame({'product_url':product_urls})
df
```

	product_url
0	https://archive.ics.uci.edu/dataset/53/iris
1	https://archive.ics.uci.edu/dataset/602/dry+be...
2	https://archive.ics.uci.edu/dataset/45/heart+d...
3	https://archive.ics.uci.edu/dataset/545/rice+c...
4	https://archive.ics.uci.edu/dataset/2/adult
5	https://archive.ics.uci.edu/dataset/850/raisin
6	https://archive.ics.uci.edu/dataset/942/rt-iot...
7	https://archive.ics.uci.edu/dataset/938/regens...
8	https://archive.ics.uci.edu/dataset/936/nation...
9	https://archive.ics.uci.edu/dataset/925/infrar...
10	https://archive.ics.uci.edu/dataset/920/jute+p...
11	https://archive.ics.uci.edu/dataset/915/differ...
12	https://archive.ics.uci.edu/dataset/53/iris
13	https://archive.ics.uci.edu/dataset/602/dry+be...
14	https://archive.ics.uci.edu/dataset/45/heart+d...
15	https://archive.ics.uci.edu/dataset/545/rice+c...
16	https://archive.ics.uci.edu/dataset/2/adult
17	https://archive.ics.uci.edu/dataset/850/raisin
18	https://archive.ics.uci.edu/dataset/942/rt-iot...
19	https://archive.ics.uci.edu/dataset/938/regens...
20	https://archive.ics.uci.edu/dataset/936/nation...
21	https://archive.ics.uci.edu/dataset/925/infrar...
22	https://archive.ics.uci.edu/dataset/920/jute+p...
23	https://archive.ics.uci.edu/dataset/915/differ...
24	https://archive.ics.uci.edu/dataset/53/iris
25	https://archive.ics.uci.edu/dataset/602/dry+be...
26	https://archive.ics.uci.edu/dataset/45/heart+d...
27	https://archive.ics.uci.edu/dataset/545/rice+c...

```
28        https://archive.ics.uci.edu/dataset/2/adult
29     https://archive.ics.uci.edu/dataset/850/raisin
30  https://archive.ics.uci.edu/dataset/942/rt-iot...
31  https://archive.ics.uci.edu/dataset/938/regens...
32  https://archive.ics.uci.edu/dataset/936/nation...
33  https://archive.ics.uci.edu/dataset/925/infrar...
34  https://archive.ics.uci.edu/dataset/920/jute+p...
35  https://archive.ics.uci.edu/dataset/915/differ...
36        https://archive.ics.uci.edu/dataset/53/iris
37  https://archive.ics.uci.edu/dataset/602/dry+be...
38  https://archive.ics.uci.edu/dataset/45/heart+d...
39  https://archive.ics.uci.edu/dataset/545/rice+c...
40        https://archive.ics.uci.edu/dataset/2/adult
41     https://archive.ics.uci.edu/dataset/850/raisin
42  https://archive.ics.uci.edu/dataset/942/rt-iot...
43  https://archive.ics.uci.edu/dataset/938/regens...
44  https://archive.ics.uci.edu/dataset/936/nation...
45  https://archive.ics.uci.edu/dataset/925/infrar...
46  https://archive.ics.uci.edu/dataset/920/jute+p...
47  https://archive.ics.uci.edu/dataset/915/differ...
48        https://archive.ics.uci.edu/dataset/53/iris
49  https://archive.ics.uci.edu/dataset/602/dry+be...
50  https://archive.ics.uci.edu/dataset/45/heart+d...
51  https://archive.ics.uci.edu/dataset/545/rice+c...
52        https://archive.ics.uci.edu/dataset/2/adult
53     https://archive.ics.uci.edu/dataset/850/raisin
54  https://archive.ics.uci.edu/dataset/942/rt-iot...
55  https://archive.ics.uci.edu/dataset/938/regens...
56  https://archive.ics.uci.edu/dataset/936/nation...
57  https://archive.ics.uci.edu/dataset/925/infrar...
58  https://archive.ics.uci.edu/dataset/920/jute+p...
59  https://archive.ics.uci.edu/dataset/915/differ...
```

```
Dataset_name=[]
Data_type=[]
Task=[]
Attribute_type=[]
No_of_instances=[]
No_of_attribute=[]
Year=[]

for url in product_urls:
    driver.get(url)
    time.sleep(2)
    try:

name_of_the_dataset=driver.find_element(By.XPATH,'/html/body/div/div[1
]/div[1]/main/div/div[1]/div[1]/div[1]/div[2]/div/h1')
        Dataset_name.append(name_of_the_dataset.text)
    except NoSuchElementException:
```

```
        Dataset_name.append('-')
    try:

dataset_type=driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1
]/main/div/div[1]/div[1]/div[2]/div[2]/div[1]/p')
        Data_type.append(dataset_type.text)
    except NoSuchElementException:
        Data_type.append('-')
    try:

task=driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/
div/div[1]/div[1]/div[2]/div[2]/div[3]/p')
        Task.append(task.text)
    except NoSuchElementException:
        Task.append('-')
    try:

attribute_type=driver.find_element(By.XPATH,'/html/body/div/div[1]/
div[1]/main/div/div[1]/div[1]/div[2]/div[2]/div[4]/p')
        Attribute_type.append(attribute_type.text)
    except NoSuchElementException:
        Attribute_type.append('-')

    try:

no_of_instances=driver.find_element(By.XPATH,'/html/body/div/div[1]/
div[1]/main/div/div[1]/div[1]/div[2]/div[2]/div[5]/p')
        No_of_instances.append(no_of_instances.text)
    except NoSuchElementException:
        No_of_instances.append('-')

    try:

no_of_attribute=driver.find_element(By.XPATH,'/html/body/div/div[1]/
div[1]/main/div/div[1]/div[1]/div[2]/div[2]/div[6]/p')
        No_of_attribute.append(no_of_attribute.text)
    except NoSuchElementException:
        No_of_attribute.append('-')

    try:

no_of_year=driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/
main/div/div[1]/div[1]/div[2]/div[1]/div/p/span')
        Year.append(no_of_year.text)
    except NoSuchElementException:
        Year.append('-')

print(len(Dataset_name),len(Data_type),len(Task),len(Attribute_type),l
en(No_of_instances),len(No_of_attribute),len(Year),len(product_urls))
```

```
60 60 60 60 60 60 60 60
```

```
import pandas as pd
df=pd.DataFrame({'data_name':Dataset_name,'data_type':Data_type,'task'
:Task,'attribute_type':Attribute_type,'no_of_instances':No_of_instance
s,'no_of_attributes':No_of_attribute,'year':
Year,'products_urls':product_urls})
df
```

```
                                    data_name  \
0                                        Iris
1                            Dry Bean Dataset
2                               Heart Disease
3                   Rice (Cammeo and Osmancik)
4                                       Adult
5                                      Raisin
6                                   RT-IoT2022
7           Regensburg Pediatric Appendicitis
8       National Poll on Healthy Aging (NPHA)
9           Infrared Thermography Temperature
10                                   Jute Pest
11  Differentiated Thyroid Cancer Recurrence
12                                       Iris
13                           Dry Bean Dataset
14                              Heart Disease
15                  Rice (Cammeo and Osmancik)
16                                      Adult
17                                     Raisin
18                                  RT-IoT2022
19          Regensburg Pediatric Appendicitis
20      National Poll on Healthy Aging (NPHA)
21          Infrared Thermography Temperature
22                                   Jute Pest
23  Differentiated Thyroid Cancer Recurrence
24                                       Iris
25                           Dry Bean Dataset
26                              Heart Disease
27                  Rice (Cammeo and Osmancik)
28                                      Adult
29                                     Raisin
30                                  RT-IoT2022
31          Regensburg Pediatric Appendicitis
32      National Poll on Healthy Aging (NPHA)
33          Infrared Thermography Temperature
34                                   Jute Pest
35  Differentiated Thyroid Cancer Recurrence
36                                       Iris
37                           Dry Bean Dataset
38                              Heart Disease
39                  Rice (Cammeo and Osmancik)
```

```
40                                      Adult
41                                     Raisin
42                                  RT-IoT2022
43          Regensburg Pediatric Appendicitis
44     National Poll on Healthy Aging (NPHA)
45          Infrared Thermography Temperature
46                                  Jute Pest
47  Differentiated Thyroid Cancer Recurrence
48                                       Iris
49                            Dry Bean Dataset
50                               Heart Disease
51                 Rice (Cammeo and Osmancik)
52                                      Adult
53                                     Raisin
54                                  RT-IoT2022
55          Regensburg Pediatric Appendicitis
56     National Poll on Healthy Aging (NPHA)
57          Infrared Thermography Temperature
58                                  Jute Pest
59  Differentiated Thyroid Cancer Recurrence

                               data_type
task  \
0                                Tabular
Classification
1                           Multivariate
Classification
2                           Multivariate
Classification
3                           Multivariate
Classification
4                           Multivariate
Classification
5                           Multivariate
Classification
6   Tabular, Sequential, Multivariate  Classification, Regression,
Clustering
7                         Tabular, Image
Classification
8                                Tabular
Classification
9                                Tabular
Regression
10                                 Image
Classification, Other
11                               Tabular
Classification
12                               Tabular
Classification
```

```
13                          Multivariate
Classification
14                          Multivariate
Classification
15                          Multivariate
Classification
16                          Multivariate
Classification
17                          Multivariate
Classification
18  Tabular, Sequential, Multivariate  Classification, Regression,
Clustering
19                       Tabular, Image
Classification
20                                Tabular
Classification
21                                Tabular
Regression
22                                  Image
Classification, Other
23                                Tabular
Classification
24                                Tabular
Classification
25                          Multivariate
Classification
26                          Multivariate
Classification
27                          Multivariate
Classification
28                          Multivariate
Classification
29                          Multivariate
Classification
30  Tabular, Sequential, Multivariate  Classification, Regression,
Clustering
31                       Tabular, Image
Classification
32                                Tabular
Classification
33                                Tabular
Regression
34                                  Image
Classification, Other
35                                Tabular
Classification
36                                Tabular
Classification
37                          Multivariate
```

```
Classification
38                        Multivariate
Classification
39                        Multivariate
Classification
40                        Multivariate
Classification
41                        Multivariate
Classification
42  Tabular, Sequential, Multivariate  Classification, Regression,
Clustering
43                      Tabular, Image
Classification
44                               Tabular
Classification
45                               Tabular
Regression
46                                 Image
Classification, Other
47                               Tabular
Classification
48                               Tabular
Classification
49                        Multivariate
Classification
50                        Multivariate
Classification
51                        Multivariate
Classification
52                        Multivariate
Classification
53                        Multivariate
Classification
54  Tabular, Sequential, Multivariate  Classification, Regression,
Clustering
55                      Tabular, Image
Classification
56                               Tabular
Classification
57                               Tabular
Regression
58                                 Image
Classification, Other
59                               Tabular
Classification

                attribute_type no_of_instances no_of_attributes  \
0                         Real             150                4
1                Integer, Real           13611               16
```

2	Categorical, Integer, Real	303	13
3	Real	3810	7
4	Categorical, Integer	48842	14
5	Real, Integer	900	7
6	Real, Categorical	123117	84
7	Real, Categorical, Integer	782	53
8	Categorical	714	14
9	Real, Categorical	1020	33
10	Categorical	7235	17
11	Real, Categorical, Integer	383	16
12	Real	150	4
13	Integer, Real	13611	16
14	Categorical, Integer, Real	303	13
15	Real	3810	7
16	Categorical, Integer	48842	14
17	Real, Integer	900	7
18	Real, Categorical	123117	84
19	Real, Categorical, Integer	782	53
20	Categorical	714	14
21	Real, Categorical	1020	33
22	Categorical	7235	17
23	Real, Categorical, Integer	383	16
24	Real	150	4
25	Integer, Real	13611	16
26	Categorical, Integer, Real	303	13
27	Real	3810	7
28	Categorical, Integer	48842	14
29	Real, Integer	900	7
30	Real, Categorical	123117	84
31	Real, Categorical, Integer	782	53
32	Categorical	714	14
33	Real, Categorical	1020	33
34	Categorical	7235	17
35	Real, Categorical, Integer	383	16
36	Real	150	4
37	Integer, Real	13611	16
38	Categorical, Integer, Real	303	13
39	Real	3810	7
40	Categorical, Integer	48842	14
41	Real, Integer	900	7
42	Real, Categorical	123117	84
43	Real, Categorical, Integer	782	53
44	Categorical	714	14
45	Real, Categorical	1020	33
46	Categorical	7235	17
47	Real, Categorical, Integer	383	16
48	Real	150	4
49	Integer, Real	13611	16
50	Categorical, Integer, Real	303	13

```
51                         Real            3810                 7
52        Categorical, Integer           48842                14
53                Real, Integer             900                 7
54           Real, Categorical          123117                84
55  Real, Categorical, Integer             782                53
56                 Categorical             714                14
57           Real, Categorical            1020                33
58                 Categorical            7235                17
59  Real, Categorical, Integer             383                16

                                                        year  \
0   A small classic dataset from Fisher, 1936. One...
1   Images of 13,611 grains of 7 different registe...
2   4 databases: Cleveland, Hungary, Switzerland, ...
3   A total of 3810 rice grain's images were taken...
4   Predict whether income exceeds $50K/yr based o...
5   Images of the Kecimen and Besni raisin varieti...
6   The RT-IoT2022, a proprietary dataset derived ...
7   This repository holds the data from a cohort o...
8   This is a subset of the NPHA dataset filtered ...
9   The Infrared Thermography Temperature Dataset ...
10  This dataset has 17 classes. Data are divided ...
11  This data set contains 13 clinicopathologic fe...
12  A small classic dataset from Fisher, 1936. One...
13  Images of 13,611 grains of 7 different registe...
14  4 databases: Cleveland, Hungary, Switzerland, ...
15  A total of 3810 rice grain's images were taken...
16  Predict whether income exceeds $50K/yr based o...
17  Images of the Kecimen and Besni raisin varieti...
18  The RT-IoT2022, a proprietary dataset derived ...
19  This repository holds the data from a cohort o...
20  This is a subset of the NPHA dataset filtered ...
21  The Infrared Thermography Temperature Dataset ...
22  This dataset has 17 classes. Data are divided ...
23  This data set contains 13 clinicopathologic fe...
24  A small classic dataset from Fisher, 1936. One...
25  Images of 13,611 grains of 7 different registe...
26  4 databases: Cleveland, Hungary, Switzerland, ...
27  A total of 3810 rice grain's images were taken...
28  Predict whether income exceeds $50K/yr based o...
29  Images of the Kecimen and Besni raisin varieti...
30  The RT-IoT2022, a proprietary dataset derived ...
31  This repository holds the data from a cohort o...
32  This is a subset of the NPHA dataset filtered ...
33  The Infrared Thermography Temperature Dataset ...
34  This dataset has 17 classes. Data are divided ...
35  This data set contains 13 clinicopathologic fe...
36  A small classic dataset from Fisher, 1936. One...
37  Images of 13,611 grains of 7 different registe...
```

```
38  4 databases: Cleveland, Hungary, Switzerland, ...
39  A total of 3810 rice grain's images were taken...
40  Predict whether income exceeds $50K/yr based o...
41  Images of the Kecimen and Besni raisin varieti...
42  The RT-IoT2022, a proprietary dataset derived ...
43  This repository holds the data from a cohort o...
44  This is a subset of the NPHA dataset filtered ...
45  The Infrared Thermography Temperature Dataset ...
46  This dataset has 17 classes. Data are divided ...
47  This data set contains 13 clinicopathologic fe...
48  A small classic dataset from Fisher, 1936. One...
49  Images of 13,611 grains of 7 different registe...
50  4 databases: Cleveland, Hungary, Switzerland, ...
51  A total of 3810 rice grain's images were taken...
52  Predict whether income exceeds $50K/yr based o...
53  Images of the Kecimen and Besni raisin varieti...
54  The RT-IoT2022, a proprietary dataset derived ...
55  This repository holds the data from a cohort o...
56  This is a subset of the NPHA dataset filtered ...
57  The Infrared Thermography Temperature Dataset ...
58  This dataset has 17 classes. Data are divided ...
59  This data set contains 13 clinicopathologic fe...
```

```
                                        products_urls
0         https://archive.ics.uci.edu/dataset/53/iris
1   https://archive.ics.uci.edu/dataset/602/dry+be...
2   https://archive.ics.uci.edu/dataset/45/heart+d...
3   https://archive.ics.uci.edu/dataset/545/rice+c...
4         https://archive.ics.uci.edu/dataset/2/adult
5      https://archive.ics.uci.edu/dataset/850/raisin
6   https://archive.ics.uci.edu/dataset/942/rt-iot...
7   https://archive.ics.uci.edu/dataset/938/regens...
8   https://archive.ics.uci.edu/dataset/936/nation...
9   https://archive.ics.uci.edu/dataset/925/infrar...
10  https://archive.ics.uci.edu/dataset/920/jute+p...
11  https://archive.ics.uci.edu/dataset/915/differ...
12        https://archive.ics.uci.edu/dataset/53/iris
13  https://archive.ics.uci.edu/dataset/602/dry+be...
14  https://archive.ics.uci.edu/dataset/45/heart+d...
15  https://archive.ics.uci.edu/dataset/545/rice+c...
16        https://archive.ics.uci.edu/dataset/2/adult
17     https://archive.ics.uci.edu/dataset/850/raisin
18  https://archive.ics.uci.edu/dataset/942/rt-iot...
19  https://archive.ics.uci.edu/dataset/938/regens...
20  https://archive.ics.uci.edu/dataset/936/nation...
21  https://archive.ics.uci.edu/dataset/925/infrar...
22  https://archive.ics.uci.edu/dataset/920/jute+p...
23  https://archive.ics.uci.edu/dataset/915/differ...
24        https://archive.ics.uci.edu/dataset/53/iris
```

```
25  https://archive.ics.uci.edu/dataset/602/dry+be...
26  https://archive.ics.uci.edu/dataset/45/heart+d...
27  https://archive.ics.uci.edu/dataset/545/rice+c...
28        https://archive.ics.uci.edu/dataset/2/adult
29     https://archive.ics.uci.edu/dataset/850/raisin
30  https://archive.ics.uci.edu/dataset/942/rt-iot...
31  https://archive.ics.uci.edu/dataset/938/regens...
32  https://archive.ics.uci.edu/dataset/936/nation...
33  https://archive.ics.uci.edu/dataset/925/infrar...
34  https://archive.ics.uci.edu/dataset/920/jute+p...
35  https://archive.ics.uci.edu/dataset/915/differ...
36        https://archive.ics.uci.edu/dataset/53/iris
37  https://archive.ics.uci.edu/dataset/602/dry+be...
38  https://archive.ics.uci.edu/dataset/45/heart+d...
39  https://archive.ics.uci.edu/dataset/545/rice+c...
40        https://archive.ics.uci.edu/dataset/2/adult
41     https://archive.ics.uci.edu/dataset/850/raisin
42  https://archive.ics.uci.edu/dataset/942/rt-iot...
43  https://archive.ics.uci.edu/dataset/938/regens...
44  https://archive.ics.uci.edu/dataset/936/nation...
45  https://archive.ics.uci.edu/dataset/925/infrar...
46  https://archive.ics.uci.edu/dataset/920/jute+p...
47  https://archive.ics.uci.edu/dataset/915/differ...
48        https://archive.ics.uci.edu/dataset/53/iris
49  https://archive.ics.uci.edu/dataset/602/dry+be...
50  https://archive.ics.uci.edu/dataset/45/heart+d...
51  https://archive.ics.uci.edu/dataset/545/rice+c...
52        https://archive.ics.uci.edu/dataset/2/adult
53     https://archive.ics.uci.edu/dataset/850/raisin
54  https://archive.ics.uci.edu/dataset/942/rt-iot...
55  https://archive.ics.uci.edu/dataset/938/regens...
56  https://archive.ics.uci.edu/dataset/936/nation...
57  https://archive.ics.uci.edu/dataset/925/infrar...
58  https://archive.ics.uci.edu/dataset/920/jute+p...
59  https://archive.ics.uci.edu/dataset/915/differ...
```