Vidyavardhaka Sangha®, Mysore
VIDYAVARDHAKA COLLEGE OF ENGINEERING
Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi
(Approved by AICTE, New Delhi & Government of Karnataka)
Accredited by NBA (CV, CS, EE, EC, IS & ME) | NAAC with 'A' Grade
P.B. No. 206, Gokulam III Stage, Mysuru-570 002, Karnataka, India
Phone: +91 821 4276201 /202 /225, Fax: +91 824 2510677
Web: http://www.vvce.ac.in @vvceofficial

Advanced Python Programming Laboratory

(BISAP317)

Activity based assessment

On

"Web Scraping"

Submitted By:

Saloni P Jain – 4VV22IS090

Vineeta Koppalkar – 4VV22IS121

Submitted To:

Prof Kasturi Rangan
Department of
Information Science and Engineering
VVCE

CONTENT

INTRODUCTION TO WEBSCRAPING

Web scraping is the process of extracting data from websites. It involves fetching the web page's HTML code and then parsing it to extract the desired information. Web scraping can be done manually, but it is often automated using software tools or programming scripts.

Here are the key steps involved in web scraping:

Sending a Request:

- Use a programming language (such as Python with libraries like Requests) or a tool to send an HTTP request to the website you want to scrape.

Fetching the Web Page:

- The server responds to the request by sending back the HTML content of the web page.

Parsing the HTML:

- Once you have the HTML content, use a parser (like BeautifulSoup in Python) to navigate through the HTML and extract the relevant data.

Selecting Elements:

- Identify the HTML elements that contain the data you want to scrape (e.g., divs, spans, tables).

Extracting Data:

- Extract the data from the selected HTML elements. This may involve extracting text, links, images, or other types of content.

Storing Data:

- Save the extracted data in a structured format, such as a CSV file, database, or other suitable storage.

MOTIVATION

Data Collection for Research:

- Researchers may use web scraping to gather data for academic or scientific research. This can include collecting information for studies in fields such as social sciences, economics, or epidemiology.

Competitive Analysis:

- Businesses may scrape data from competitor websites to analyze pricing strategies, product offerings, customer reviews, and other competitive intelligence.

Market Research:

- Web scraping can be used for market research to analyze trends, consumer behaviour, and product reviews across various websites.

Lead Generation:

- Sales and marketing professionals may scrape websites to gather leads and contact information for potential customers.

Monitoring Public Opinion:

- Social media scraping can be employed to monitor public sentiment, track trends, and analyze discussions on various platforms.

Job Market Analysis:

- Job seekers or human resources professionals might use web scraping to analyze job market trends, salary information, and job postings.

Weather Data Collection:

- Meteorologists and weather enthusiasts may scrape data from various sources to analyze and predict weather patterns.

Government and Public Data Analysis:

- Researchers and analysts may use web scraping to gather public data from government websites for analysis and reporting.

PROBLEM STATEMENT

Extracting specific data from websites manually is time-consuming and tedious, hence we use web-scraping, which automatically extracts data from webpages.

The Key Requirements are as follows:

1. **Website Selection:** Identify and select a list of e-commerce websites relevant for scraping. Ensure compliance with the legal and ethical aspects of web scraping.

2. **Product Selection:** Allow users to specify the products or product categories for which they want to monitor prices.

3. **Real-time Scraping:** Implement a real-time scraping mechanism to regularly fetch and update pricing information from selected websites.

4. **Data Accuracy:** Ensure accurate extraction of pricing details, including discounts, promotions, and any other relevant information.

5. **User Interface:** Develop a user friendly interface that allows user to manage their preferences

DESIGN OF THE CODE

1. Making a Request

```
[*]: import pandas as pd
     import requests
     from bs4 import BeautifulSoup
     product_name=[]
     reviews=[]
     links=[]
     for i in range(2,10):
         url="https://www.flipkart.com/search?q=apple+mobiles&sid=tyy%2C4io&as=on&as-show=on&otracker=AS_QueryStore_Orgar
         r=requests.get(url)
         print(r)
```

```
<Response [200]>
```

In simple terms, this code uses Python to fetch the content of a webpage and print it out. It shows the response status code to confirm if the request was successful, and then prints the webpage's content, which includes its HTML code.

2. Extracting URLs

This code sends a request to a webpage, then prints the URL of the request and the status code of the response.

3. Installing a Beautiful Soup Library

This code fetches a web page, prints its status code, then uses BeautifulSoup to parse the HTML content and prints it in a prettified format

4. Extracting multiple page’s data

This code extracts data from a website’s page which has all the relevant information about products, links and its reviews

5. Extracting product names, reviews and links

This code fetches a web page, parses its HTML content using BeautifulSoup, then finds the div tag and then the class of it in which product names and reviews are displayed. Then we look for anchor tag and href attribute where the product linked are stored

6. Save data to CSV

This script fetches a web page, parses its HTML content using BeautifulSoup, finds all div tags with class 'head', extracts the text from these tags, and stores the title details in a list of dictionaries. Finally, it writes the title details to a CSV file named 'titles.csv'.

MODIFICATION

The user is provided with 6 different choices of input , from which the user can go with his top 2 choices and then datasets of the selected choices will be displayed. The average is calculated of each input, based on reviews scraped. Finally the best of two will be displayed to the user so that the user can go with the best choice.

IMPLEMENTATION

Code of Web scraping:

```
import pandas as pd import
requests from bs4 import
BeautifulSoup
product_name=[] reviews=[]
links=[] for i in range(2,10):

url="https://www.flipkart.com/search?q=apple+mobiles&sid=tyy%2C4io&as=on&as-show=on&otracker=AS_QueryStore_OrganicAutoSuggest_1_7_na_na_na&otracker1=AS_QueryStore_OrganicAutoSuggest_1_7_na_na_na&as-pos=1&as-type=RECENT&suggestionId=apple+mobiles%7 CMobiles&requestId=74513d15-9fcc-41ea-a7e0-16499e689ee3&as-backfill=on&otracker=nmen u_sub_Electronics_0_Apple&page="+str(i)
r=requests.get(url)

  soup=BeautifulSoup(r.text,"lxml")
  box=soup.find("div",class_="_1YokD2 _3Mn1Gg")
  names=box.find_all("div",class_="_4rR01T") for i
  in names: name=i.text
     product_name.append(name)

  reviews_data=box.find_all("div",class_="_3LWZlK")
  for i in reviews_data: name=i.text
  reviews.append(name)
  content=BeautifulSoup(r.content,'html.parser')
  data=content.find_all("div",{'class':'_2kHMtA'})
  start_link="https://www.flipkart.com" for items
  in data:
```

```
rest_link=items.find('a')['href']
links.append(start_link+rest_link)
df=pd.DataFrame({"Product name":product_name,"Links of products":links,"Reviews":reviews})
df.to_csv("C:/Users/prafu/OneDrive/Desktop/python aba datasets/Apple_mobile.csv")
print(df)
```

OUTPUT

	Product name	Links of products	Reviews
0	Apple iPhone 15 Pro (Natural Titanium, 256 GB)	https://www.flipkart.com/apple-iphone-15-pro-n...	4.7
1	Apple iPhone 12 (Blue, 128 GB)	https://www.flipkart.com/apple-iphone-12-blue-...	4.6
2	Apple iPhone 15 Pro Max (Blue Titanium, 256 GB)	https://www.flipkart.com/apple-iphone-15-pro-m...	4.6
3	Apple iPhone 15 Pro Max (White Titanium, 256 GB)	https://www.flipkart.com/apple-iphone-15-pro-m...	4.6
4	Apple iPhone 15 Pro Max (Natural Titanium, 256...	https://www.flipkart.com/apple-iphone-15-pro-m...	4.6
..	...	...	...
187	Apple iPhone 12 mini (Purple, 64 GB)	https://www.flipkart.com/apple-iphone-12-mini-...	4.5
188	Apple iPhone 13 Pro Max (Alpine Green, 256 GB)	https://www.flipkart.com/apple-iphone-13-pro-m...	4.6
189	Apple iPhone 13 Pro Max (Sierra Blue, 512 GB)	https://www.flipkart.com/apple-iphone-13-pro-m...	4.6
190	Apple iPhone XR (Yellow, 64 GB) (Includes EarP...	https://www.flipkart.com/apple-iphone-xr-yello...	4.6
191	Apple iPhone XR (Yellow, 128 GB) (Includes Ear...	https://www.flipkart.com/apple-iphone-xr-yello...	4.6

[192 rows x 3 columns]

Modification code:

```
import pandas as pd
datasets = {
    '1': 'Apple_mobiles_with_links.csv',
    '2': 'Lenovo_mobiles.csv',
    '3': 'Mi_mobiles.csv',
    '4': 'oppo_mobiles.csv',
```

```
    '5': 'Poco_mobiles.csv',
    '6': 'realme_mobiles.csv'
}


def display_dataset_names():
    print("Available datasets:") for
    key, value in datasets.items():
    print(f"{key}: {value}")

def display_dataset(choice):
    filename = datasets.get(choice)
    if filename:
        df = pd.read_csv(filename)
        print(f"\nDataset {choice} - {datasets[choice]}")
        print(df)
        average_review = df['Reviews'].mean()
    print("Average Review Score:", average_review)
    return average_review else:
        print("Invalid choice. Please choose a number between 1 and 6.")

def main():
    display_dataset_names()

    print("\nSelect the first dataset number to compare (1-6):")
    dataset_choice_1 = input() if dataset_choice_1 not in
    datasets:
        print("Invalid dataset number. Please choose a number between 1 and 6.")
        return
    avg_review_1 = display_dataset(dataset_choice_1)
```

```
    print("\nSelect the second dataset number to compare (1-6):")
    dataset_choice_2 = input() if dataset_choice_2 not in
    datasets:
        print("Invalid dataset number. Please choose a number between 1 and 6.")
        return
    avg_review_2 = display_dataset(dataset_choice_2)

    if avg_review_1 > avg_review_2: print(f"\nDataset
        {dataset_choice_1} has the best reviews.")
    elif avg_review_1 < avg_review_2:
        print(f"\nDataset {dataset_choice_2} has the best reviews.")
    else: print("\nBoth datasets have the same average review
        score.")

if __name__ == "__main__":
    main()
```

OUTPUT

```
Available datasets:
1: Apple_mobiles_with_links.csv
2: Lenovo_mobiles.csv
3: Mi_mobiles.csv
4: oppo_mobiles.csv
5: Poco_mobiles.csv
6: realme_mobiles.csv

Select the first dataset number to compare (1-6):
 1

Dataset 1 - Apple_mobiles_with_links.csv
     Unnamed: 0                                       Product name  \
0             0                    Apple iPhone 12 (Green, 64 GB)
1             1                    Apple iPhone 12 (Black, 64 GB)
2             2                Apple iPhone 14 (Starlight, 256 GB)
3             3                     Apple iPhone 15 (Pink, 256 GB)
4             4  Apple iPhone 15 Pro Max (Natural Titanium, 256...
..          ...                                                ...
187         187      Apple iPhone 13 Pro Max (Sierra Blue, 512 GB)
188         188             Apple iPhone XR ((PRODUCT)RED, 128 GB)
189         189      Apple iPhone 13 Pro Max (Sierra Blue, 256 GB)
190         190  Apple iPhone XR (White, 128 GB) (Includes EarP...
191         191  Apple iPhone XR (Blue, 128 GB) (Includes EarPo...
```

```
                                  Links of products  Reviews
0    https://www.flipkart.com/apple-iphone-12-green...      4.6
1    https://www.flipkart.com/apple-iphone-12-black...      4.6
2    https://www.flipkart.com/apple-iphone-14-starl...      4.6
3    https://www.flipkart.com/apple-iphone-15-pink-...      4.6
4    https://www.flipkart.com/apple-iphone-15-pro-m...      4.6
..                                                 ...      ...
187  https://www.flipkart.com/apple-iphone-13-pro-m...      4.6
188  https://www.flipkart.com/apple-iphone-xr-produ...      4.6
189  https://www.flipkart.com/apple-iphone-13-pro-m...      4.6
190  https://www.flipkart.com/apple-iphone-xr-white...      4.6
191  https://www.flipkart.com/apple-iphone-xr-blue-...      4.6

[192 rows x 4 columns]
Average Review Score: 4.570312500000009

Select the second dataset number to compare (1-6):
 3

Dataset 3 - Mi_mobiles.csv
     Unnamed: 0                              Product name  \
0             0  REDMI Note 11 Pro (Stealth Black, 128 GB)
1             1                  Redmi 9 (Sky Blue, 64 GB)
2             2           Mi 11 Lite (Vinyl Black, 128 GB)
3             3             Redmi 8 (Sapphire Blue, 64 GB)
4             4          Redmi K20 Pro (Flame Red, 128 GB)
..          ...                                        ...
187         187         11 Lite NE (Tuscany Coral, 128 GB)
188         188        11 Lite NE (Diamond Dazzle, 128 GB)
189         189           11 Lite NE (Vinyl Black, 128 GB)
190         190                     Redmi Y2 (Gold, 32 GB)
191         191                      Redmi 2 (White, 8 GB)

                                  Links of products  Reviews
0    https://www.flipkart.com/redmi-note-11-pro-ste...      4.1
1    https://www.flipkart.com/redmi-9-sky-blue-64-g...      4.3
2    https://www.flipkart.com/mi-11-lite-vinyl-blac...      4.2
3    https://www.flipkart.com/redmi-8-sapphire-blue...      4.4
4    https://www.flipkart.com/redmi-k20-pro-flame-r...      4.5
..                                                 ...      ...
187  https://www.flipkart.com/11-lite-ne-tuscany-co...      3.7
188  https://www.flipkart.com/11-lite-ne-diamond-da...      3.7
189  https://www.flipkart.com/11-lite-ne-vinyl-blac...      3.7
190  https://www.flipkart.com/redmi-y2-gold-32-gb/p...      4.5
191  https://www.flipkart.com/redmi-2-white-8-gb/p/...      4.2

[192 rows x 4 columns]
Average Review Score: 4.259895833333336

Dataset 1 has the best reviews.
```

Line plot code:

```
import pandas as pd

ap=pd.read_csv('Apple_mobiles_with_links.csv')

lp=pd.read_csv('Lenovo_mobiles.csv')

mp=pd.read_csv('Mi_mobiles.csv')

op=pd.read_csv('oppo_mobiles.csv')
```

```
pp=pd.read_csv('Poco_mobiles.csv')
rp=pd.read_csv('realme_mobiles.csv')
print('Average reviews')
r1=ap['Reviews'].mean()
print(r1)
r2=lp['Reviews'].mean()
print(r2)
r3=mp['Reviews'].mean()
print(r3)
r4=op['Reviews'].mean()
print(r4)
r5=pp['Reviews'].mean()
print(r5)
r6=rp['Reviews'].mean()
print(r6)
```

OUTPUT

```
Average reviews
4.570312500000009
3.991463414634147
4.259895833333336
4.32374999999999
4.272727272727275
4.33999999999999
```

```
data = {
   'Brand_name': ['Apple', 'Lenovo', 'Mi', 'oppo','Poco','realme'],
   'Reviews': [4.5703, 3.9914, 4.2598, 4.3237,4.2727,4.3399],


}

# Create a DataFrame from the dictionary
df = pd.DataFrame(data)
df.to_csv("C:/Users/prafu/OneDrive/Desktop/python aba datasets/brands_reviewa.csv")
print(df)
```

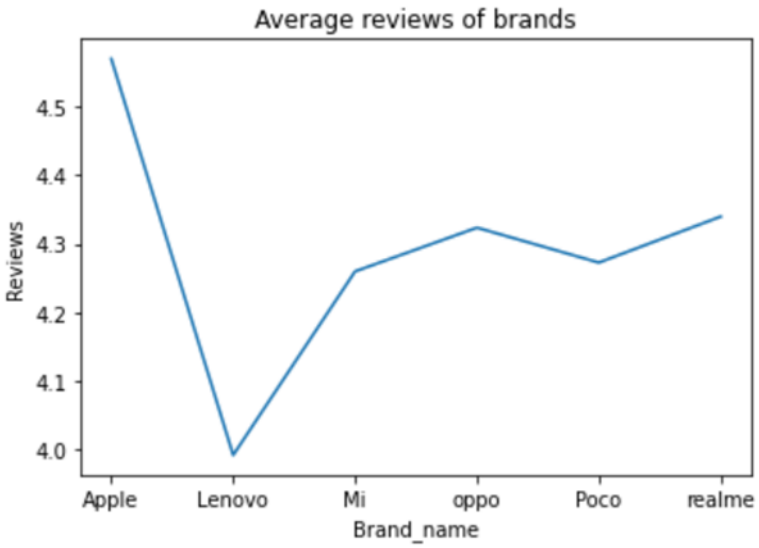
OUTPUT

	Brand_name	Reviews
0	Apple	4.5703
1	Lenovo	3.9914
2	Mi	4.2598
3	oppo	4.3237
4	Poco	4.2727
5	realme	4.3399

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
rv=pd.read_csv('brands_reviewa.csv')
sns.lineplot(data=df, x='Brand_name', y='Reviews')
plt.title('Average reviews of brands')
```

OUTPUT

```
Text(0.5, 1.0, 'Average reviews of brands')
```



CONCLUSION

In conclusion, web scraping is a powerful technique for extracting valuable data from websites, enabling individuals and organizations to gain insights, make informed decisions, and automate various processes. The process involves sending HTTP requests to retrieve HTML content, parsing the HTML to extract relevant information, and storing the data for analysis or other purposes.

Web scraping finds applications across various domains, including research, business intelligence, market analysis, and more. However, it is essential to approach web scraping responsibly and ethically. Scrappers should respect the terms of service of the websites being scraped, adhere to legal regulations, and avoid causing harm or disruption to the targeted websites.

Common tools and libraries, such as BeautifulSoup, Requests, Scrapy, and Selenium, facilitate the web scraping process, providing developers with the necessary functionality to interact with and extract data from websites.

As web scraping continues to evolve, challenges like handling dynamic content, anti-scraping mechanisms, and ethical considerations remain crucial areas of consideration. Navigating these challenges requires a thoughtful and responsible approach to ensure the sustainability and legality of web scraping activities.

In conclusion, when conducted ethically and within legal boundaries, web scraping can be a valuable tool for acquiring data, driving innovation, and gaining a competitive edge in various fields.