NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

Department of Artificial Intelligence & Machine Learning

Academic Year 2022-23(ODD)

Report

for

Mini project-III (20AIM59A)

On

"WEB CRAWLER USING BIG DATA"

By

Name	USN
R. Roopam Chowdary	1NH20AI085
C. Rohith Kumar	1NH20AI124
P. Saketh Sree Ram	1NH20AI075

Under the Guidance of

Ms. Jimsha

Assistant professor,

Dept. of Artificial Intelligence & Machine Learning,

New Horizon College of Engineering,

Bangalore-560103

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

Department of Artificial Intelligence & Machine Learning

CERTIFICATE

Certified that the Mini Project- III with the subject code 20AIM59A work entitled **"WEB CRAWLER USING BIG DATA"** carried out by R. Roopam Chowdary, usn :1NH20AI085, C.Rohith Kumar,usn:1NH20AI124 and P.Saketh Sree Ram, usn:1NH20AI075. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of Mini Project work.

Prof. Jimsha	**Dr. N V Uma Reddy**
Internal Guide	Head of Department

External Viva
Examiner **Signature with date**

1.

2.

ABSTRACT

Testing theoretical models against trustworthy and sizable databases is essential in the social and economic sciences. The main research problem is to create a cost-effective, well-structured database that is suitable for the specified research topic. This paper focuses on crawler algorithms that have demonstrated their efficacy as a data base development tool in a variety of challenging situations. In order to map business interactions utilizing social media information sources, a sophisticated research procedure is first explained and shown. In this instance, we show how data collection from search robots may be utilized to characterize business interactions in a specific setting by mapping complicated network linkages. The scenario is then expanded, and a framework for three fundamentally distinct research models is then shown, including exploration, classification, and time series analysis, where crawler programs may be successfully used. When no prior statistical information was known on the activities of the Hungarian web agency business, we report our findings in the case of exploration. We demonstrate how the most popular e-business model categories may be created using the most popular Hungarian web names. In the third study, we employed a crawler to collect data from a single website on the values of real, pre-defined records with low-cost plane ticket prices. We highlight key conceptual findings and potential of crawler-based research in e-business based on the experiences.

Table of Contents

Chapter **Page No.**

List of Figures

CHAPTER-1

INTRODUCTION

1.1 Introduction

The World Wide Web (WWW) uses a client-server model for the internet. It is a strong system built on the server's total autonomy for providing internet-based content. The data is organized in a massive, dispersed, and non-linear text structure called a hypertext document system. These systems specify a document's hypertextual content as text or picture fragments that are connected to other documents by anchor tags. A standardized method of accessing and presenting hyperlinked documents is provided by HTTP and HTML. Search engines are used by web browsers to look across servers for the necessary information pages. The client side processes the pages that the servers send. Using the internet to get information from the World Wide Web has become a crucial component of modern life. Approximately 7.049 billion people live on the planet as of today, 2.40 billion of whom (34.3%) utilize the Internet (see Figure 1.1). Internet users climbed from.36 billion in 2000 to 2.40 billion in 2012, a rise of 566.4% from that year to that one. Out of 3.92 billion people in Asia, 1.076 billion (or 27.5%) utilize the Internet, compared to.137 billion (11.4%) in India, which has 1.2 billion people. Future expansion is anticipated to continue at the same pace, and it won't be long until people begin to feel that life would be incomplete without the Internet.

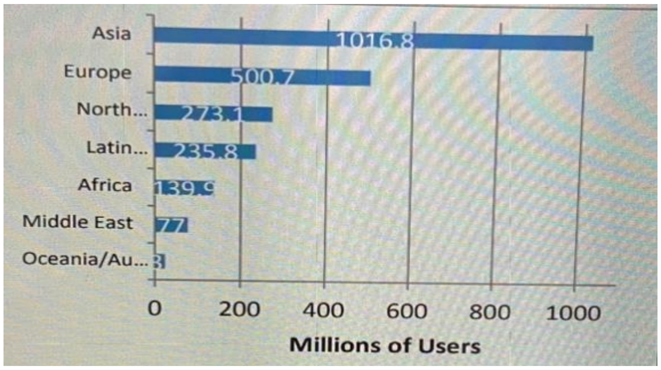


Figure 1.1: shows the distribution of Internet users worldwide by geographical area

The extent of the World Wide Web has increased dramatically since 1990. There are currently thought to be roughly 55 billion publicly indexable online documents [4] scattered throughout the globe on hundreds of servers. Finding information in such a vast database of online documents that are accessible over the WWW is difficult. Even though users know where to go for content by knowing its URLs, it is uncertain if they will be able to obtain it because the Web is always evolving. The following three categories are used to group information retrieval tools:

a) Web Crawling/Scraping
b) Meta search engines
c) Search engines

Web scraping/crawling is a technique used to extract data from websites. It involves sending a request to a website's server and then parsing the HTML or XML response to extract the desired information. This technique is commonly used by businesses, researchers, and individuals to collect data for a variety of purposes such as market research, price comparison, and sentiment analysis.

1.2 Objective

A search engine bot, web crawler, or spider downloads and indexes material from all across the Internet. Such a bot aims to learn the subjects of (nearly) all web pages so that it can obtain the information as needed. They are referred to as "web crawlers" since automatically accessing a website and retrieving data using software constitutes the technical word for crawling.

web crawling and web scraping is to automatically collect and extract data from websites This allows search engines to give relevant links in response to user search queries (or another search engine).

The objective of this project is to extract specific information from a website. In this we are using amazon website to get specific search query information like price, ratings, ratings count etc.. and these are used to analysis and visualization of the query.

1.3 Literature Survey

Survey Title	Reference	Methodology	Outcome/Results	Remarks
"Scrapy: A Fast and Powerful Scraping and Web Crawling Framework"	Author: Pablo Hoffman	Introduces the Scrapy web crawling and scraping library and discusses more complex issues including managing cookies, logging into websites, and integrating Scrapy with other Python libraries.	A thorough introduction to web collecting and indexing by using Scrapy library	Scraping a website too frequently can put a strain on the website servers.
"Web Scraping with Python and Beautiful Soup"	Author: Ryan Mitchell	A step-by-step manual for using the Beautiful Soup library to scrape the web. explains the fundamentals of HTML and CSS and demonstrates how to utilize Beautiful Soup to extract information from web pages.	It provides information on how to use the Beautiful Soup library to extract data from web pages.	These techniques are not accurate of data and contain errors.
"Data Crawling and Processing with Apache Nutch and Apache Tika"	Author: Lewis John McGibbney	An overview of how to use the Apache Tika and Nutch libraries for data processing and web crawling. explains the fundamentals of utilizing Nutch and Tika and demonstrates how to use them to collect data from websites and utilize it in Hadoop processing.	A demonstration on how to use the Nutch and Tika libraries to collect data from website urls and analyze it in Hadoop.	Sometimes these contain null values which effect in analyzing data.
"Web scraping with Python: A practical guide"	Author: Ahmed Rafik	It explains the fundamentals of online scraping and demonstrates how to collect data from web	It gives brief basics of scraping data from web pages using python libraries.	It should be done frequently because of many updates in websites.

		pages using Python packages like Requests, Beautiful Soup, and Selenium.		
"A survey on web scraping techniques"	Author: S.S. Bhowmick	A study of the methods for web scraping currently in use. This article discusses several webs scraping strategies, including DOM parsing, regular expressions, and machine learning-based methods, as well as the difficulties and drawbacks of each methodology.	A detailed examination of the difficulties and restrictions associated with current web scraping methods.	Some websites don’t allow the scraping or crawling.
"Web scraping using Python: A beginner's guide"	Author: A. K. Singh	A step-by-step manual for Python web scraping for beginners. explains the fundamentals of web scraping and demonstrates how to collect data from online pages using Python tools like Requests and Beautiful Soup.	Web scraping using tools like requests and Beautifulsoup and their outcomes.	Web scraping and web crawling can raise ethical concerns as it can be used for malicious purposes such as identity theft, phishing, and spreading malware.
“Data Processing and Analysis with MongoDB and Apache Spark”	Author: A.C. Kayi	It discusses the fundamentals of MongoDB and Spark and demonstrates how to utilize them for complex data processing and analysis activities as well as storing and querying massive volumes of data.	It results in handling and analyzing huge data with Apache Spark and MongoDB.	Slow performance for large datasets. Spark need more skills to work with it.

1.3 Existing system

Thousands of links and sub-links are hidden on websites. Links on websites often lead to wealth of knowledge and information. This study focuses primarily on the challenge of anticipating the growth of a web structure early in the Website Development Life Cycle (WDLC), particularly during planning and demand collection. Finding a suitable tool to help developers with these stages is lacking. The goal of this study is to quantify a website's logical size in order to estimate the development process' duration and cost depending on the content and internal organization of the website.

1.4 Proposed system

A system for obtaining, storing, and analyzing online pages is a web crawler, commonly referred to as a spider. It accomplishes the task of arranging web pages so that consumers can quickly obtain the information they want. This is accomplished by gathering a few websites and using links to access fresh material. Although there are many uses for web crawlers, they are most famous for being a crucial part of search engines. These search engines gather a number of web pages, index them, and then enable users to look for pages that correspond to their query.

The project is a web scraping script that extracts data from Amazon's website for a specific search query and saves the data in a CSV file. The program uses the requests and BeautifulSoup libraries to scrape data from the website and extract information such as the product name, rating, rating count, price, and product URL for each item returned in the search results. The program also utilizes the plotly library to create various visualizations of the data such as line plots, scatter plots, bar plots, histograms, and box plots.

CHAPTER-2

SYSTEM REQUIREMENTS

2.1 Hardware requirements

- Hard disk minimum of 40 GB
- RAM minimum of 4 GB

2.2 Software requirements

- Python 3.x software
- PyCharm
- Internet connection is required in order to run the crawler.
- Python Libraries: Pyspark, requests, bs4-BeautifulSoup, pandas, matplotlib.pyplot

CHAPTER-3

SYSTEM DESIGN

3.1 System architecture

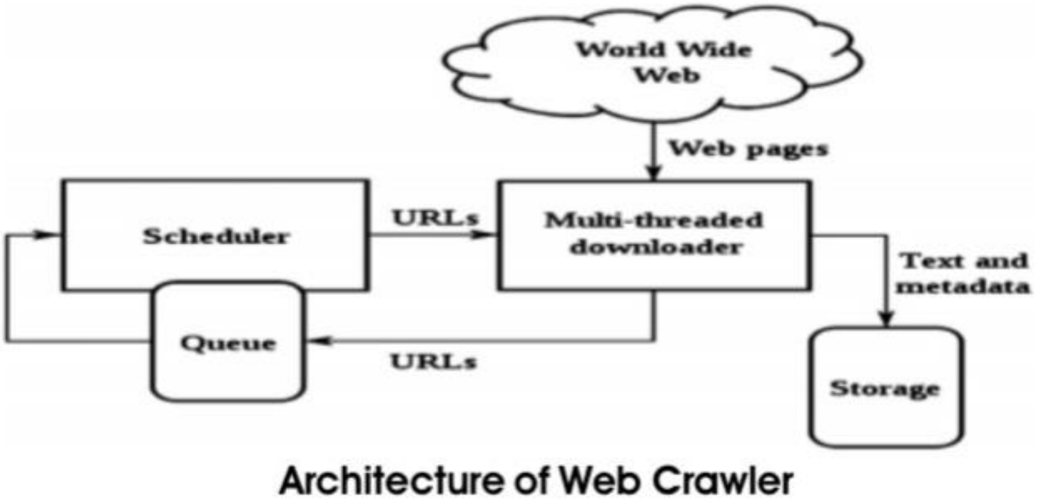


Figure 3.1: Architecture of web Crawler

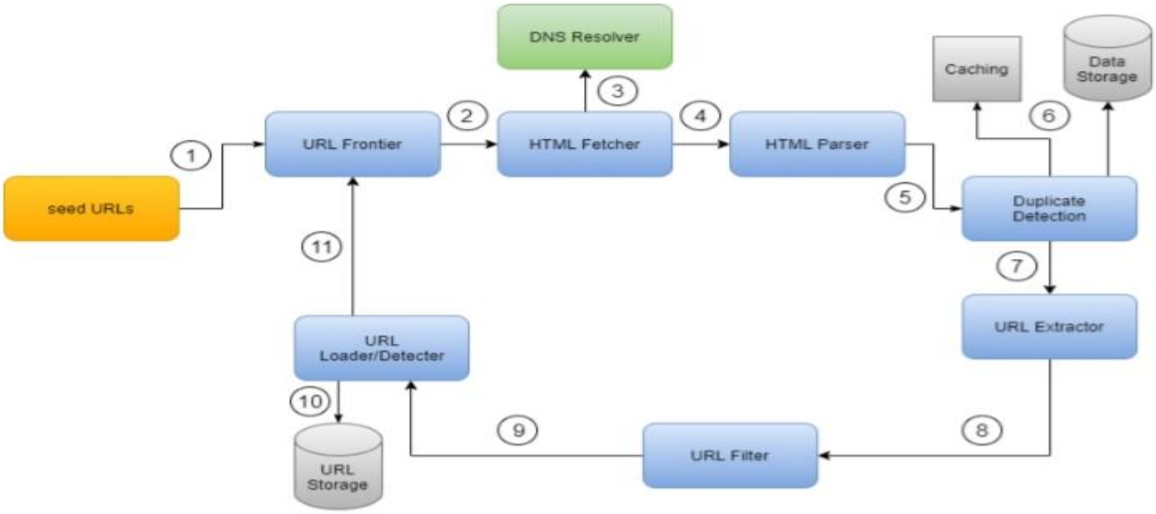
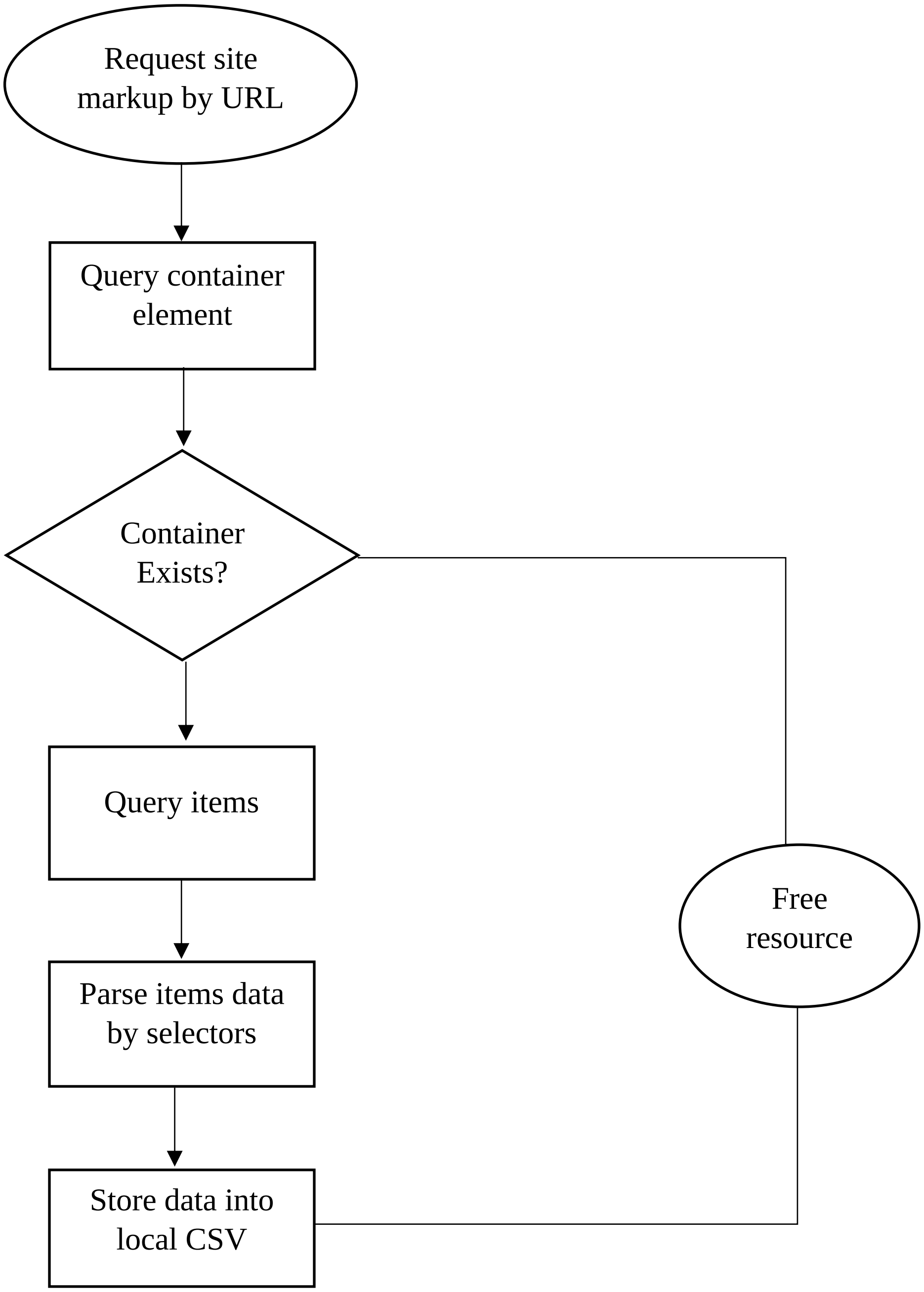


Figure 3.2: Working of web crawler

3.2 Algorithms/ Flow charts

Flowchart:


Request site markup by URL
Query container element
Container Exists?
Query items
Parse items data by selectors
Store data into local CSV
Free resource

Algorithm:

1. The script starts by defining the necessary headers for the requests and setting the search query to "Lenovo".
2. The script then uses a for loop to iterate through multiple pages of search results.
3. For each iteration, the script uses the requests library to send a GET request to the URL of the current page of the search results.
4. The response received from the website is then passed to the BeautifulSoup library, which is used to parse the HTML and extract the desired data such as product name, rating, rating count, price, and product URL.
5. The script then uses the extracted data to create a pandas dataframe.
6. The dataframe is then saved as a CSV file.
7. The script then uses the plotly library to create various visualizations of the data such as line plots, scatter plots, bar plots, histograms, and box plots.
8. Additionally, the script uses matplotlib and seaborn to create additional visualizations.
9. Finally, the script uses the pyspark library to create a Spark session and load the data into a Spark dataframe for further analysis.

CHAPTER-4

IMPLEMENTATION

4.1 Pseudocode

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
from time import sleep
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0) Gecko/20100101 Firefox/94.0',
    'Accept-Language': 'en-US, en;q=0.5'
}

search_query = 'lenovo'.replace(' ', '+')
base_url = 'https://www.amazon.com/s?k={0}'.format(search_query)
items = []
for i in range(1, 5):
    print('Processing {0}...'.format(base_url + '&page={0}'.format(i)))
    response = requests.get(base_url + '&page={0}'.format(i), headers=headers)
    soup = BeautifulSoup(response.content, 'html.parser')

    results = soup.find_all('div', {'class': 's-result-item', 'data-component-type': 's-search-result'})

    for result in results:
        product_name = result.h2.text

        try:
            rating = result.find('i', {'class': 'a-icon'}).text
            rating_count = result.find_all('span', {'aria-label': True})[1].text
        except AttributeError:
            continue

        try:
            price_element = result.select('span.a-price span.a-offscreen')
            if price_element:
                price = price_element[0].text
                price = float(price.replace('$','').replace(',',''))
            else:
                price = None
            product_url = 'https://amazon.com' + result.h2.a['href']
            # print(rating_count, product_url)
            items.append([product_name, rating, rating_count, price, product_url])
        except AttributeError:
            continue

df = pd.DataFrame(items, columns=['product', 'rating', 'rating count', 'price', 'product url'])
df.to_csv('{0}.csv'.format(search_query), index=False)
import plotly.express as px
data = pd.read_csv("lenovo.csv")
fig = px.line(data_frame=data, x='product', y='price')
fig.show()
fig = px.scatter(data_frame=data, x='rating', y='rating count')
fig.show()
fig = px.bar(data_frame=data, x='product', y='rating count', color='rating')
fig.show()
fig = px.histogram(data_frame=data, x='rating count')
fig.show()
```

```
fig = px.imshow(data.corr(), labels={"x": "rating count", "y": "price"})
fig.show()
fig = px.box(data_frame=data, y='rating count')
fig.show()
df.hist('price',bins=20)
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))
plt.hist(df["price"])
plt.show()
df["price"].value_counts().plot(kind='bar')
plt.figure(figsize=(12,6))
plt.show()
plt.scatter(df["rating count"], df["price"])
plt.show()
df.boxplot(column=["price"])
plt.show()
import seaborn as sns
sns.pairplot(df)
plt.show()
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("Analysis").getOrCreate()
data = spark.read.format("csv").option("header", "true").load("dell.csv")
```

The above program is a web scraping script that is designed to extract data from Amazon's website for a specific search query. The script uses the requests and BeautifulSoup libraries to scrape data from the website and extract information such as the product name, rating, rating count, price, and product URL for each item returned in the search results.

The script starts by defining the necessary headers for the requests and setting the search query to "Lenovo". The headers are used to identify the script as a web browser and provide information about the language preferences of the user.

Then, the script uses a for loop to iterate through multiple pages of search results and extract the data for each item on each page. For each iteration, the script uses the requests library to send a GET request to the URL of the current page of the search results. The response received from the website is then passed to the BeautifulSoup library, which is used to parse the HTML and extract the desired data.

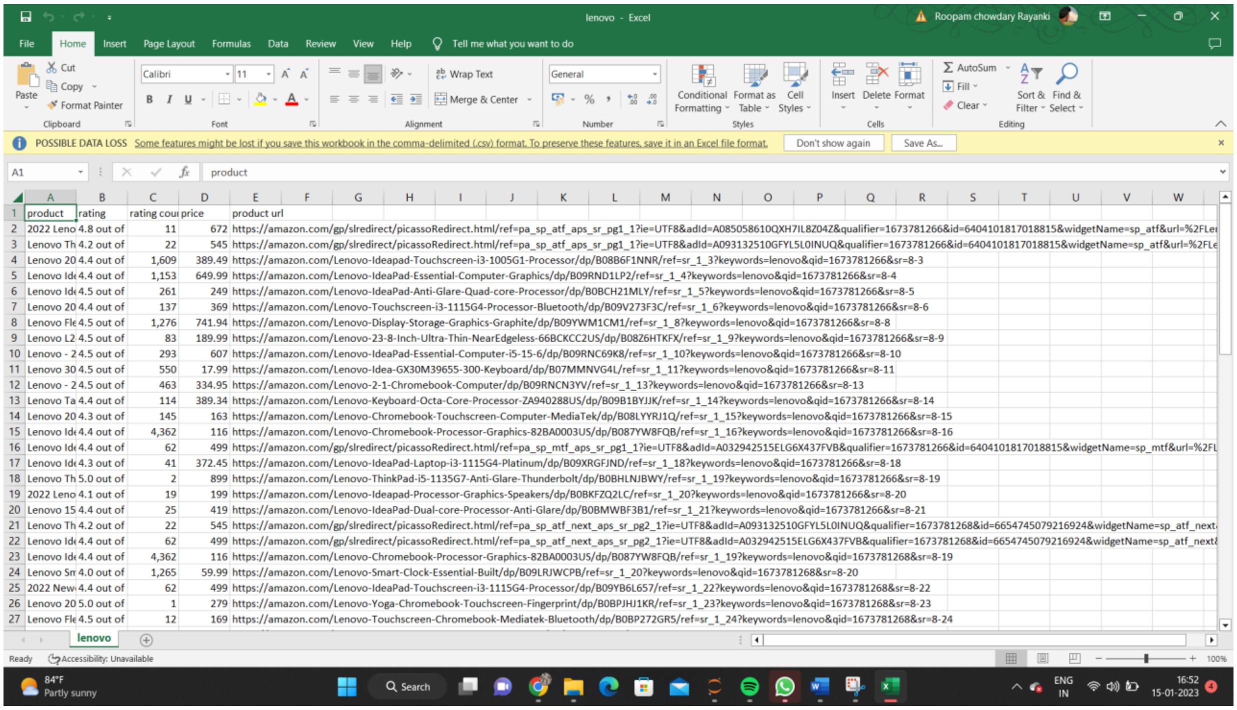
The script then uses the extracted data to create a pandas dataframe. The dataframe is then saved as a CSV file, this file can be used for further analysis.

The script also uses the plotly library to create various visualizations of the data. These visualizations include line plots, scatter plots, bar plots, histograms, and box plots. These visualizations allow for a quick and easy way to analyze the data and identify trends or patterns in the data. Additionally, the script uses matplotlib and seaborn to create additional visualizations such as histograms and scatter plots.

Finally, the script uses the pyspark library to create a Spark session and load the data into a Spark dataframe for further analysis. This allows for more advanced analysis of the data using Spark's built-in functions and tools.

Overall, this script is a useful tool for extracting and analyzing data from Amazon's website. It allows for quick and easy data extraction and visualization, and also provides the ability to perform more advanced analysis using Spark.

4.2 Result


lenovo - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Help
Tell me what you want to do
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.
Don't show again
Save As...
A1
product
lenovo
Ready
Accessibility: Unavailable

product	rating	rating coui	price	product url
2022 Leno	4.8 out of	11	672	https://amazon.com/gp/slredirect/picassoRedirect.html/ref=pa_sp_atf_aps_sr_pg1_1?ie=UTF8&adId=A085058610QXH7IL8Z04Z&qualifier=1673781266&id=6404101817018815&widgetName=sp_atf&url=%2FLe
Lenovo Th	4.2 out of	22	545	https://amazon.com/gp/slredirect/picassoRedirect.html/ref=pa_sp_atf_aps_sr_pg1_1?ie=UTF8&adId=A093132510GFYL5L0INUQ&qualifier=1673781266&id=6404101817018815&widgetName=sp_atf&url=%2FLe
Lenovo 20	4.4 out of	1,609	389.49	https://amazon.com/Lenovo-Ideapad-Touchscreen-i3-1005G1-Processor/dp/B08B6F1NNR/ref=sr_1_3?keywords=lenovo&qid=1673781266&sr=8-3
Lenovo Id	4.4 out of	1,153	649.99	https://amazon.com/Lenovo-IdeaPad-Essential-Computer-Graphics/dp/B09RND1LP2/ref=sr_1_4?keywords=lenovo&qid=1673781266&sr=8-4
Lenovo Id	4.5 out of	261	249	https://amazon.com/Lenovo-IdeaPad-Anti-Glare-Quad-core-Processor/dp/B0BCH21MLY/ref=sr_1_5?keywords=lenovo&qid=1673781266&sr=8-5
Lenovo 20	4.4 out of	137	369	https://amazon.com/Lenovo-Touchscreen-i3-1115G4-Processor-Bluetooth/dp/B09V273F3C/ref=sr_1_6?keywords=lenovo&qid=1673781266&sr=8-6
Lenovo Fle	4.5 out of	1,276	741.94	https://amazon.com/Lenovo-Display-Storage-Graphics-Graphite/dp/B09YWM1CM1/ref=sr_1_8?keywords=lenovo&qid=1673781266&sr=8-8
Lenovo L2	4.5 out of	83	189.99	https://amazon.com/Lenovo-23-8-Inch-Ultra-Thin-NearEdgeless-66BCKCC2US/dp/B08Z6HTKFX/ref=sr_1_9?keywords=lenovo&qid=1673781266&sr=8-9
Lenovo - 2	4.5 out of	293	607	https://amazon.com/Lenovo-IdeaPad-Essential-Computer-i5-15-6/dp/B09RNC69K8/ref=sr_1_10?keywords=lenovo&qid=1673781266&sr=8-10
Lenovo 30	4.5 out of	550	17.99	https://amazon.com/Lenovo-Idea-GX30M39655-300-Keyboard/dp/B07MMNVG4L/ref=sr_1_11?keywords=lenovo&qid=1673781266&sr=8-11
Lenovo - 2	4.5 out of	463	334.95	https://amazon.com/Lenovo-2-1-Chromebook-Computer/dp/B09RNCN3YV/ref=sr_1_13?keywords=lenovo&qid=1673781266&sr=8-13
Lenovo Ta	4.4 out of	114	389.34	https://amazon.com/Lenovo-Keyboard-Octa-Core-Processor-ZA940288US/dp/B09B1BYJJK/ref=sr_1_14?keywords=lenovo&qid=1673781266&sr=8-14
Lenovo 20	4.3 out of	145	163	https://amazon.com/Lenovo-Chromebook-Touchscreen-Computer-MediaTek/dp/B08LYYRJ1Q/ref=sr_1_15?keywords=lenovo&qid=1673781266&sr=8-15
Lenovo Id	4.4 out of	4,362	116	https://amazon.com/Lenovo-Chromebook-Processor-Graphics-82BA0003US/dp/B087YW8FQB/ref=sr_1_16?keywords=lenovo&qid=1673781266&sr=8-16
Lenovo Id	4.4 out of	62	499	https://amazon.com/gp/slredirect/picassoRedirect.html/ref=pa_sp_mtf_aps_sr_pg1_1?ie=UTF8&adId=A032942515ELG6X437FVB&qualifier=1673781266&id=6404101817018815&widgetName=sp_mtf&url=%2FL
Lenovo Id	4.3 out of	41	372.45	https://amazon.com/Lenovo-IdeaPad-Laptop-i3-1115G4-Platinum/dp/B09XRGFJND/ref=sr_1_18?keywords=lenovo&qid=1673781266&sr=8-18
Lenovo Th	5.0 out of	2	899	https://amazon.com/Lenovo-ThinkPad-i5-1135G7-Anti-Glare-Thunderbolt/dp/B0BHLNJBWY/ref=sr_1_19?keywords=lenovo&qid=1673781266&sr=8-19
2022 Leno	4.1 out of	19	199	https://amazon.com/Lenovo-Ideapad-Processor-Graphics-Speakers/dp/B0BKFZQ2LC/ref=sr_1_20?keywords=lenovo&qid=1673781266&sr=8-20
Lenovo 15	4.4 out of	25	419	https://amazon.com/Lenovo-IdeaPad-Dual-core-Processor-Anti-Glare/dp/B0BMWBF3B1/ref=sr_1_21?keywords=lenovo&qid=1673781266&sr=8-21
Lenovo Th	4.2 out of	22	545	https://amazon.com/gp/slredirect/picassoRedirect.html/ref=pa_sp_atf_next_aps_sr_pg2_1?ie=UTF8&adId=A093132510GFYL5L0INUQ&qualifier=1673781268&id=6654745079216924&widgetName=sp_atf_next
Lenovo Id	4.4 out of	62	499	https://amazon.com/gp/slredirect/picassoRedirect.html/ref=pa_sp_atf_next_aps_sr_pg2_1?ie=UTF8&adId=A032942515ELG6X437FVB&qualifier=1673781268&id=6654745079216924&widgetName=sp_atf_next&
Lenovo Id	4.4 out of	4,362	116	https://amazon.com/Lenovo-Chromebook-Processor-Graphics-82BA0003US/dp/B087YW8FQB/ref=sr_1_19?keywords=lenovo&qid=1673781268&sr=8-19
Lenovo Sm	4.0 out of	1,265	59.99	https://amazon.com/Lenovo-Smart-Clock-Essential-Built/dp/B09LRJWCPB/ref=sr_1_20?keywords=lenovo&qid=1673781268&sr=8-20
2022 New	4.4 out of	62	499	https://amazon.com/Lenovo-IdeaPad-Touchscreen-i3-1115G4-Processor/dp/B09YB6L657/ref=sr_1_22?keywords=lenovo&qid=1673781268&sr=8-22
Lenovo 20	5.0 out of	1	279	https://amazon.com/Lenovo-Yoga-Chromebook-Touchscreen-Fingerprint/dp/B0BPJHJ1KR/ref=sr_1_23?keywords=lenovo&qid=1673781268&sr=8-23
Lenovo Fle	4.5 out of	12	169	https://amazon.com/Lenovo-Touchscreen-Chromebook-Mediatek-Bluetooth/dp/B0BP272GR5/ref=sr_1_24?keywords=lenovo&qid=1673781268&sr=8-24

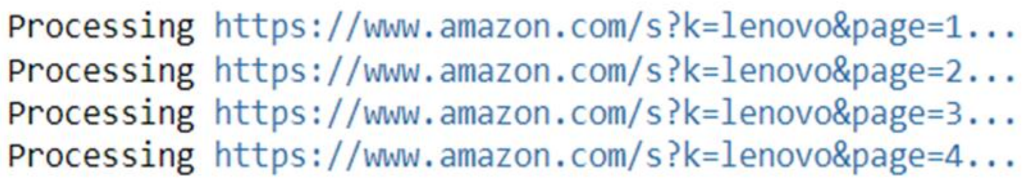
Figure 4.1: CSV sheet of search query of 'lenovo"

The results of the above program would be a CSV file containing the extracted data from the Amazon website for the search query "lenovo". The file would contain columns for the product name, rating, rating count, price, and product URL for each item returned in the search results.

Additionally, the script would create various visualizations of the data using the plotly library, such as line plots, scatter plots, bar plots, histograms, and box plots. These visualizations would allow for a quick and easy way to analyze the data and identify trends or patterns in the data. Additionally, the script would use matplotlib and seaborn to create additional visualizations such as histograms and scatter plots.

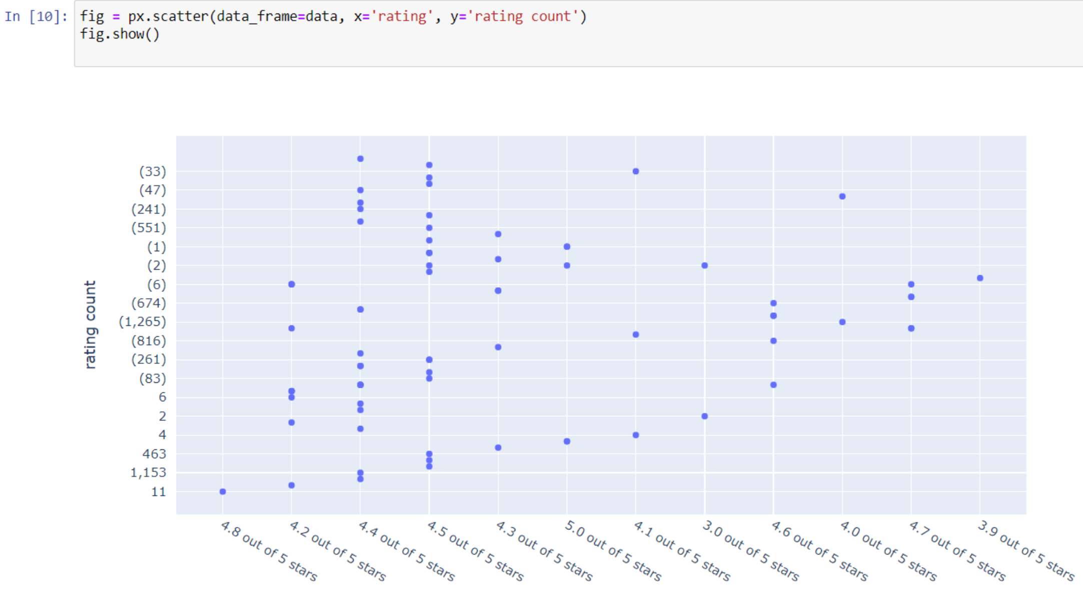
Finally, the script would also create a Spark dataframe containing the extracted data, this dataframe can be further used for advanced analysis with the help of Spark's built-in functions and tools.

Overall, the results of the above program would be a comprehensive set of data and visualizations that could be used to gain insights into the products, pricing, and customer reviews for the search query "lenovo" on Amazon's website.



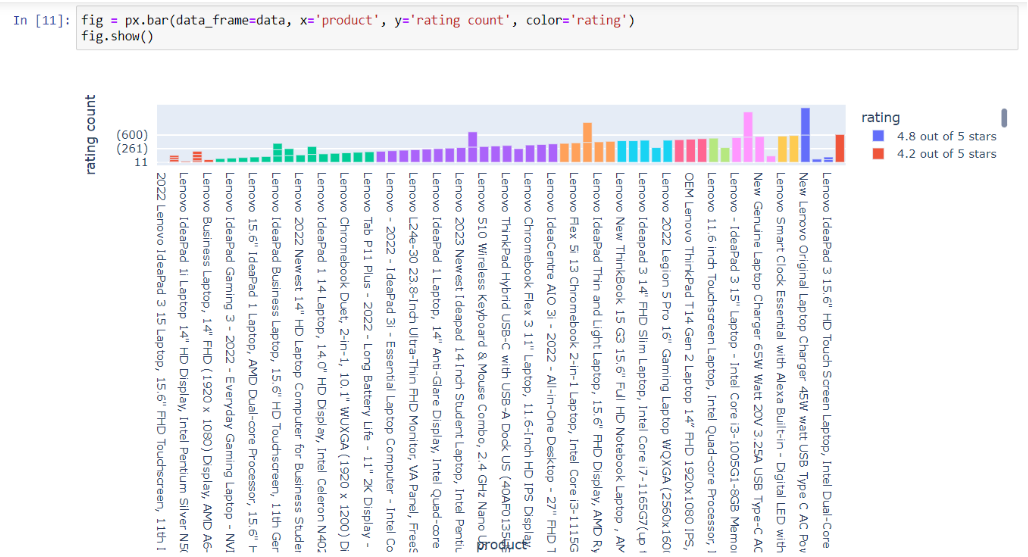
```
Processing https://www.amazon.com/s?k=lenovo&page=1...
Processing https://www.amazon.com/s?k=lenovo&page=2...
Processing https://www.amazon.com/s?k=lenovo&page=3...
Processing https://www.amazon.com/s?k=lenovo&page=4...
```

Figure 4.2: Web pages crawling on amazon website



```
In [10]: fig = px.scatter(data_frame=data, x='rating', y='rating count')
         fig.show()
```

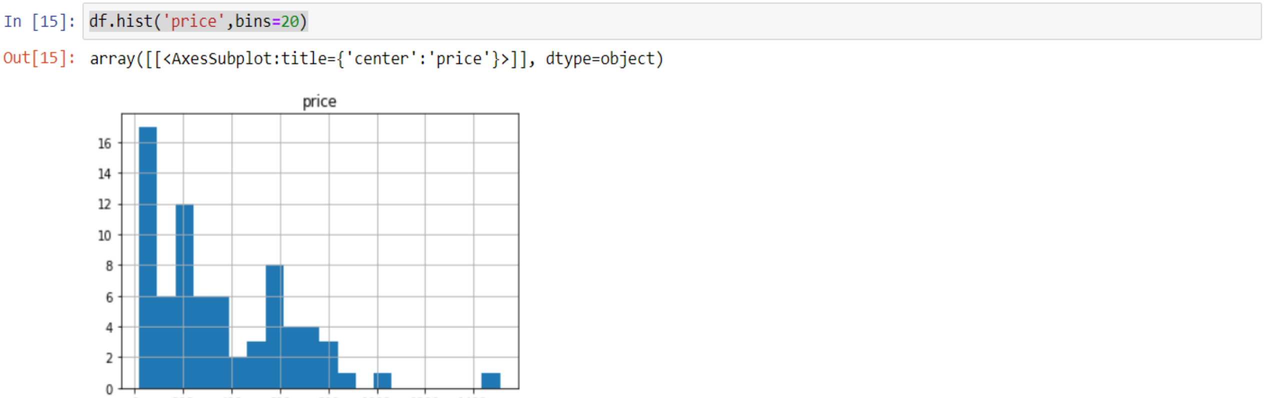
Figure 4.3: Scatterplot (x=’rating’, y=’rating count’)



```
In [11]: fig = px.bar(data_frame=data, x='product', y='rating count', color='rating')
         fig.show()
```

rating count
(600)
(261)
11
product
2022 Lenovo IdeaPad 3 15 Laptop, 15.6" FHD Touchscreen, 11th I
Lenovo IdeaPad 1i Laptop 14" HD Display, Intel Pentium Silver N5(
Lenovo Business Laptop, 14" FHD (1920 x 1080) Display, AMD A6-
Lenovo IdeaPad Gaming 3 - 2022 - Everyday Gaming Laptop - NVI
Lenovo 15.6" IdeaPad 1 Laptop, AMD Dual-core Processor, 15.6" H
Lenovo IdeaPad Business Laptop, 15.6" HD Touchscreen, 11th Ger
Lenovo 2022 Newest 14" HD Laptop Computer for Business Studer
Lenovo IdeaPad 1 14 Laptop, 14.0" HD Display, Intel Celeron N402
Lenovo Chromebook Duet, 2-in-1, 10.1" WUXGA (1920 x 1200) Di
Lenovo Tab P11 Plus - 2022 - Long Battery Life - 11" 2K Display -
Lenovo - 2022 - IdeaPad 3i - Essential Laptop Computer - Intel Co
Lenovo L24e-30 23.8-Inch Ultra-Thin FHD Monitor, VA Panel, Frees
Lenovo IdeaPad 1 Laptop, 14" Anti-Glare Display, Intel Quad-core
Lenovo 2023 Newest Ideapad 14 Inch Student Laptop, Intel Pentiu
Lenovo 510 Wireless Keyboard & Mouse Combo, 2.4 GHz Nano U
Lenovo ThinkPad Hybrid USB-C with USB-A Dock US (40AF0135US
Lenovo Chromebook Flex 3 11" Laptop, 11.6-Inch HD IPS Display
Lenovo IdeaCentre AIO 3i - 2022 - All-in-One Desktop - 27" FHD T
Lenovo Flex 5i 13 Chromebook 2-in-1 Laptop, Intel Core i3-1115G
Lenovo IdeaPad Thin and Light Laptop, 15.6" FHD Display, AMD Ry
Lenovo New ThinkBook 15 G3 15.6" Full HD Notebook Laptop , AV
Lenovo Ideapad 3 14" FHD Slim Laptop, Intel Core i7-1165G7(up t
Lenovo 2022 Legion 5 Pro 16" Gaming Laptop WQXGA (2560x160(
OEM Lenovo ThinkPad T14 Gen 2 Laptop 14" FHD 1920x1080 IPS,
Lenovo 11.6 inch Touchscreen Laptop, Intel Quad-core Processor, I
Lenovo - IdeaPad 3 15" Laptop - Intel Core i3-1005G1-8GB Memo
New Genuine Laptop Charger 65W Watt 20V 3.25A USB Type-C AC
Lenovo Smart Clock Essential with Alexa Built-in - Digital LED with
New Lenovo Original Laptop Charger 45W watt USB Type C AC Pov
Lenovo IdeaPad 3 15.6" HD Touch Screen Laptop, Intel Dual-Core
rating
4.8 out of 5 stars
4.2 out of 5 stars

Figure 4.4: Bar Plot



```
In [15]: df.hist('price',bins=20)
Out[15]: array([[<AxesSubplot:title={'center':'price'}>]], dtype=object)
```

Figure 4.5: Histogram (Price)

CHAPTER-5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

In terms of discussions, the script can be used as a starting point to extract and analyze data from other e-commerce websites as well. The script could be modified to scrape data on a regular basis, to provide an up-to-date analysis of the market trends. Additionally, the script could be enhanced to include machine learning capabilities to analyze the data and make predictions or recommendations based on the results.

Overall, the results and visualizations provided by the script can be used to gain insights into the products, pricing, and customer reviews for the search query "Lenovo" on Amazon's website. The script could be enhanced to scrape data from multiple websites, adding machine learning capabilities and scheduling the script to run on regular basis to provide an up-to-date analysis of the market trends.

Each search engine's use of web crawlers is crucial. They must deliver good performance because they are the fundamental part of all online services. The modification of data by web crawlers is extensive. It's not difficult to create an efficient web crawler to serve a variety of objectives, but the correct tactics and an efficient architecture will enable the creation of a highly intelligent web crawler program. The search engines employ a variety of crawling algorithms. For improved outcomes and high performance, a decent crawling algorithm should be used.

In conclusion, the above program is a useful tool for extracting and analyzing data from Amazon's website for a specific search query. It allows for quick and easy data extraction and visualization, and also provides the ability to perform more advanced analysis using Spark. The resulting data and visualizations can be used to gain insights into the products, pricing, and customer reviews for the search query "Lenovo" on Amazon's website.

5.2 Future Enhancement

There are several potential enhancements that could be made to the program in the future. One enhancement could be to add functionality to scrape data from multiple websites, such as other e-commerce platforms or review websites, to gather a more comprehensive dataset. Another enhancement could be to add machine learning capabilities to the program to analyze the data and make predictions or recommendations based on the results. Additionally, the script could be adjusted to scrape data on a regular basis, to provide an up-to-date analysis of the market trends.

REFERENCES

[1] "Scrapy: A Fast and Powerful Scraping and Web Crawling Framework" by Pablo Hoffman, is a comprehensive introduction to using the Scrapy library for web scraping and crawling." *Forensic Science International* 322 (2021): 110753

[2] "Web Scraping with Python and Beautiful Soup" by Ryan Mitchell, is a beginner-friendly guide to using the Beautiful Soup library for web scraping. " *International Journal of Advances in Soft Computing & Its Applications* 13, no. 3 (2021)

[3] "Data Crawling and Processing with Apache Nutch and Apache Tika" by Lewis John McGibbney, is a guide to using the Apache Nutch and Apache Tika libraries for web crawling and data processing. ." *SoftwareX* 6 (2017): 98-106.

[4] "Web scraping with Python: A practical guide" by Ahmed Rafik is a comprehensive tutorial on web scraping using Python." In *Researching cybercrimes*, pp. 435-456. Palgrave Macmillan, Cham, 2021.

[5] "A survey on web scraping techniques" by S.S. Bhowmick , A comprehensive survey of existing web scraping techniques. Covers various web scraping methods such as DOM parsing, regular expressions, and machine learning-based techniques, as well as the challenges and limitations of each method. pp. 450-454. IEEE, 2019.

[6] "Web scraping using Python: A beginner's guide" by A. K. Singh, is a beginner-friendly guide to web scraping using Python (2021) IJMI 150.

[7] Data Processing and Analysis with MongoDB and Apache Spark" by A.C. Kayi, is a guide to using MongoDB and Apache Spark for big data processing and analysis. pp. 689-693. IEEE, 2016