Python Web Scraping

Jani Begam Zahir Hussain

—

November 20, 2024

—

Data Analysis Tools Analytics

Web Scraping- Amazon Best Seller Books

Objective

The main objective of this project is to perform web scraping of the Amazon Best Seller Books Webpage using Beautifulsoup tool and compare the processing time with other web scraping tools.

Steps in performing the Process:

1. Importing all the necessary libraries
2. Web Scraping
3. Storing the scraped data into CSV file.
4. In Exploratory Data Analysis (EDA), the data cleaning and NaN value handling.
5. Creating Insights and Visualizations
6. Comparing the performance of other Web Scraping Tools.

Step:1 Importing libraries

The following libraries were used:

- **requests:** To send HTTP requests to the Amazon website.
- **BeautifulSoup:** For parsing and scraping data from the webpage.
- **csv:** To save the scraped data into a CSV file.
- **pandas:** For data manipulation and analysis.
- **matplotlib and bokeh:** For visualizations.

Step: 2 Web Scraping

- **Base URL:**
  https://www.amazon.in/gp/bestsellers/books/ref=zg_bs_pg_

- **Methodology:**
  - ✓ We sent HTTP requests to the first two pages of the Amazon best-sellers list.
  - ✓ This parsed the response HTML using BeautifulSoup to extract book details: Title, Author, Rating, Customers Rated, and Price.
  - ✓ Handled missing data by assigning default values.

- **Performance:** The scraper efficiently fetched details of 60 books due to structural complexities in the HTML of the Amazon Website.

- **Code Explanation:**

  - ✓ Finding the book elements : This line retrieves all such elements and stores them in book_elements, to process each book's details individually.

```
import requests
from bs4 import BeautifulSoup
import csv # Import the csv module

# Function to scrape a single Amazon page
def scrape_amazon_page(soup, books):
    book_elements = soup.find_all('div', class_='_cDEzb_iveVideoWrapper_JJ34T')
```

  - ✓ Extracting the Title, Author, Rating:
    - The title and author are contained in a nested , but the Rating is nested in . get_text(strip=True) retrieves the text content.
    - If author and title are missing, it defaults to "Unknown ".
    - The function .split()[0] extracts only the numeric part ("4.5" from "4.5 out of 5 stars").
    - If no rating is found, "0" is used as a fallback value.

```
for book_element in book_elements:
    # Title
    title = book_element.find('div', class_='_cDEzb_p13n-sc-css-line-clamp-1_1Fn1y')
    title = title.get_text(strip=True) if title else "Unknown Title"

    # Author
    author = book_element.find('div', class_='a-row a-size-small')
    author = author.get_text(strip=True) if author else "Unknown Author"

    # Rating
    rating = book_element.find('span', class_='a-icon-alt')
    rating = rating.get_text(strip=True).split()[0] if rating else "0"
```

- ✓ Extracting the Number of Customer Ratings:
  - The number of customer ratings is nested under a with the class 'a-icon-row'.
  - get_text(strip=True) fetches the raw number, and replace(',', '') removes commas for conversion to an integer using int().
  - If the element or value is not found, "Not Rated" is assigned.

```
# Extract the customer ratings
  customer_rating_element = book_element.find('div', class_='a-icon-row')  # Ensure the parent is 'a-icon
  if customer_rating_element:
    customers_rated = customer_rating_element.find('span', class_='a-size-small')
    if customers_rated:
      # Remove commas and convert to integer
        customers_rated = int(customers_rated.get_text(strip=True).replace(',', ''))
    else:
        customers_rated = "Not Rated"  # Use "Not Rated" if not found
  else:
      customers_rated = "Not Rated"  # Use "Not Rated" if the element is missing
```

- ✓ Extracting the Price:
  - The function get_text(strip=True) extracts the price as text (e.g., "₹1,234.00").
  - The function replace('₹', '').replace(',', '') removes the currency symbol and commas.
  - The function .split('.')[0] removes the fractional part.
  - The price is then converted to an integer using int().
  - If the price element is missing, "Price Unavailable" is assigned.

```
  if price_element:
        # Get the price text and clean it
    price = price_element.get_text(strip=True).replace('₹', '').replace(',', '').split('.')[0]
    price = int(price)  # Convert to integer
  else:
    price = "Price Unavailable"  # Handle missing price
```

- ✓ The extracted books details are appended to the books[].

```
# Append to books list
books.append({
    'title': title,
    'author': author,
    'rating': rating,
    'customers_rated': customers_rated,
    'price': price
})
```

- ✓ Using Beautifulsoup:
  - This script fetches the first two pages of Amazon India's bestsellers in the books category.
  - It constructs the URL dynamically for each page, sends an HTTP GET request with custom headers to mimic a browser, and parses the HTML response using BeautifulSoup.
  - The scrape_amazon_page function is then called to extract and append book details to the books list.

```
# Base URL and headers
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36'
}

base_url = "https://www.amazon.in/gp/bestsellers/books"

# List to store book details
books = []

# Loop through the first two pages
for page_number in range(1, 3):
    url = f"{base_url}/ref=zg_bs_pg_{page_number}_books?ie=UTF8&pg={page_number}"
    print(f"Scraping Page {page_number}: {url}")
    response = requests.get(url, headers=headers)
    soup = BeautifulSoup(response.text, 'html.parser')

    # Scrape page
    scrape_amazon_page(soup, books)

# Check results
print(f"Total Books Scraped: {len(books)}")
for book in books[:5]:  # Display first 5 books
    print(f"Title: {book['title']}")
    print(f"Author: {book['author']}")
    print(f"Rating: {book['rating']}")
    print(f"Customers Rated: {book['customers_rated']}")
    print(f"Price: {book['price']}")
    print("-" * 20)
```

- **Output:**

```
Scraping Page 1: https://www.amazon.in/gp/bestsellers/books/ref=zg_bs_pg_1_books?ie=UTF8&pg=1
Scraping Page 2: https://www.amazon.in/gp/bestsellers/books/ref=zg_bs_pg_2_books?ie=UTF8&pg=2
Total Books Scraped: 60
Title: The Psychology of Money
Author: Morgan Housel
Rating: 4.6
Customers Rated: 67648
Price: 312
--------------------
Title: Amma Diarylo Konni Pageelu
Author: Ravi Mantri
Rating: 4.8
Customers Rated: 1115
Price: 220
--------------------
Title: The Satvic Revolution: 7 Life-Changing Habits to Discover Peak Health and Joy
Author: Subah Saraf
Rating: 4.8
Customers Rated: 981
Price: 323
--------------------
Title: My First Library: Boxset of 10 Board Books for Kids
Author: Wonder House Books
Rating: 4.5
Customers Rated: 80744
Price: 399
--------------------
Title: Atomic Habits
Author: James Clear
Rating: 4.6
Customers Rated: 98774
Price: 549
--------------------
```

Step: 3 Storing the scraped data into CSV file

The scraped data was stored in a CSV file (amazon_bestsellers.csv) using the csv module.

- **Code lines:**

```python
# # Save the data to a CSV file with UTF-8 BOM encoding
with open('amazon_bestsellersfina11.csv', 'w', newline='', encoding='utf-8-sig') as csvfile:
    fieldnames = ['title', 'author', 'rating', 'customers_rated', 'price']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()
    for book in books:
        writer.writerow(book)

print(f"Data scraped and saved to 'amazon_bestsellers.csv'")
```

```
Data scraped and saved to 'amazon_bestsellers.csv'
```

Step:4 In Exploratory Data Analysis (EDA), the data cleaning and NaN value handling.

- **Code Explanation:**
  - ✓ Creating DataFrame: The books list (containing dictionaries of book details) is converted into a pandas DataFrame for structured tabular representation.
  - ✓ Checking for NaN values: It uses df.isnull().sum() to count missing (NaN) values in each column, helping identify data quality issues.
  - ✓ Dropping the NaN Rows: Optionally, rows with missing values are removed using df.dropna() (depending on whether missing values are acceptable).
  - ✓ Viewing Cleaned Data: Finally, it prints the first few rows of the cleaned dataset to confirm successful data handling.

- **Code lines:**

```
import pandas as pd

# Create a DataFrame from the scraped data
df = pd.DataFrame(books)

# Check for NaN values
print("NaN counts per column:\n", df.isnull().sum())

# Drop rows with NaN values (optional, depends on requirement)
df = df.dropna()

# Check the cleaned dataset
print("DataFrame after handling NaNs:")
print(df.head())
```

```
NaN counts per column:
 title              0
author             0
rating             0
customers_rated    0
price              0
dtype: int64
DataFrame after handling NaNs:
                                               title             author  \
0                                The Psychology of Money      Morgan Housel
1                            Amma Diarylo Konni Pageelu        Ravi Mantri
2  The Satvic Revolution: 7 Life-Changing Habits ...         Subah Saraf
3  My First Library: Boxset of 10 Board Books for...  Wonder House Books
4                                      Atomic Habits        James Clear

   rating  customers_rated  price
0     4.6            67648    312
1     4.8             1115    220
2     4.8              981    323
3     4.5            80744    399
4     4.6            98774    549
```

Step:5 Creating Insights and Visualizations

- **Question 1: List the Authors Highest Priced Book (i.e., based on price): show your result (at least top 25 highest priced book) as a data frame as well a Bar diagram.**

  - ✓ The DataFrame is sorted in descending order of the price column using sort_values('price', ascending=False).
  - ✓ The head(25) function extracts the top 25 rows with the highest prices.
  - ✓ It prints a subset of the columns (author, title, and price) for these top 25 books, allowing easy review of the most expensive books in the dataset.

```
[ ]  import matplotlib.pyplot as plt

     # Sort the DataFrame by price in descending order
     top_25_books = df.sort_values('price', ascending=False).head(25)

     # Display the top 25 highest-priced books
     print("Top 25 Highest Priced Books:")
     print(top_25_books[['author', 'title', 'price']])
```

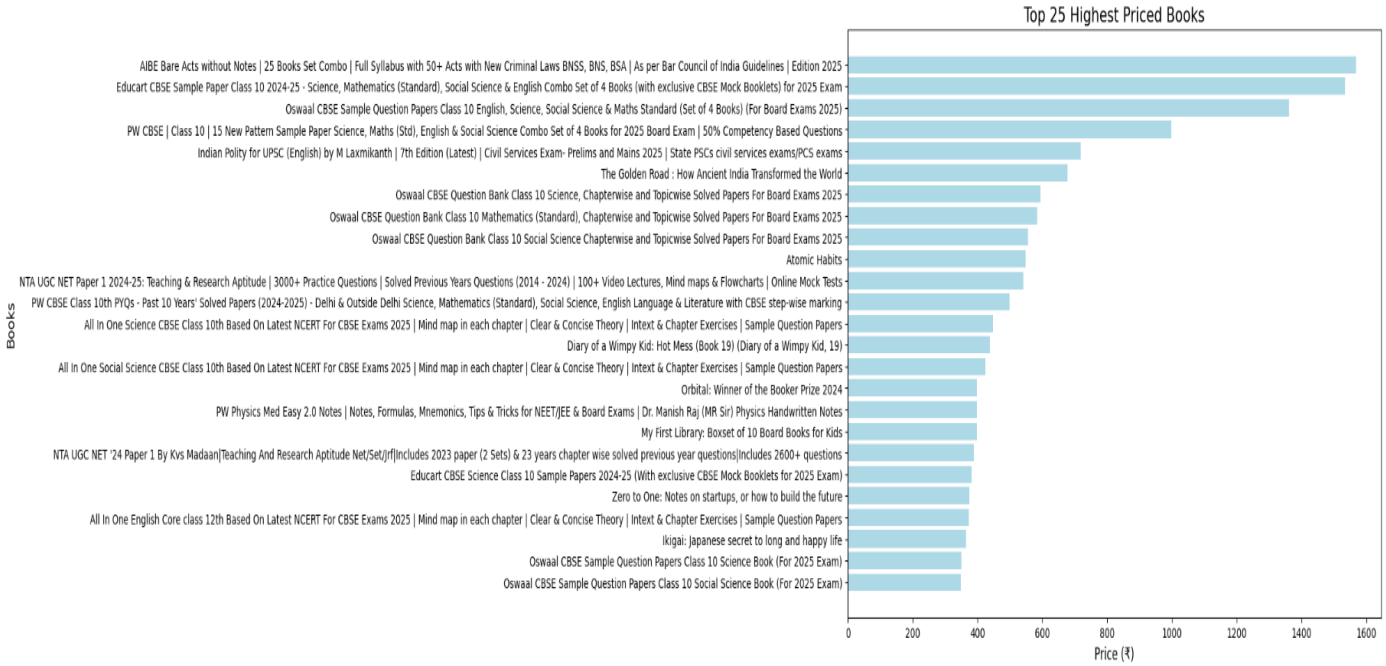
- ✓ This code visualizes the top 25 highest-priced books with a horizontal bar chart by initializing a figure with figsize = (12, 8) to set the dimensions of the chart.
- ✓ A horizontal bar chart (barh) is created with book titles on the y-axis and their prices on the x-axis. The bars are styled in lightblue.
- ✓ Adds xlabel for price in rupees (₹) and then adds ylabel to list the books and finally adds a chart title for context.
- ✓ The invert_yaxis() ensures the most expensive books appear at the top.
- ✓ The function tight_layout() adjusts spacing for better readability.
- ✓ The function plt.show() renders the chart for viewing.

```
# Plot the results
plt.figure(figsize=(12, 8))
plt.barh(top_25_books['title'], top_25_books['price'], color='lightblue')
plt.xlabel('Price (₹)', fontsize=12)
plt.ylabel('Books', fontsize=12)
plt.title('Top 25 Highest Priced Books', fontsize=16)
plt.gca().invert_yaxis()  # Invert y-axis for better readability
plt.tight_layout()
plt.show()
```

- **Output:**

```
Top 25 Highest Priced Books:
                                    author  \
58                                Lawmann's
15                                  Educart
11                  Oswaal Editorial Board
8                      PW (Physics Wallah)
34                            M Laxmikanth
38                          Wlliam Dalrymple
25                  Oswaal Editorial Board
57                  Oswaal Editorial Board
53                  Oswaal Editorial Board
4                              James Clear
36                            ARPITA KARWA
20                                      PW
44  Dr. parul Goel Ranjan , Abhshek Gupta
19                               Jeff Kinney
47                Susmita Dhar Kriti Arora
40                         Samantha Harvey
33                Dr. Manish Raj (MR. Sir)
3                       Wonder House Books
56                                KVS Madaan
17                             Prashant Kirad
50              Peter Masters, Blake,Thiel
54             Prerna Kain Srishti Agarwal
12                       Francesc Miralles
16                  Oswaal Editorial Board
49                  Oswaal Editorial Board

                                                title  price
58  AIBE Bare Acts without Notes | 25 Books Set Co...   1568
15  Educart CBSE Sample Paper Class 10 2024-25 - S...   1535
11  Oswaal CBSE Sample Question Papers Class 10 En...   1362
8   PW CBSE | Class 10 | 15 New Pattern Sample Pap...    999
34  Indian Polity for UPSC (English) by M Laxmikan...    719
38  The Golden Road : How Ancient India Transforme...    678
25  Oswaal CBSE Question Bank Class 10 Science, Ch...    594
57  Oswaal CBSE Question Bank Class 10 Mathematics...    585
53  Oswaal CBSE Question Bank Class 10 Social Scie...    556
4                                        Atomic Habits    549
36  NTA UGC NET Paper 1 2024-25: Teaching & Resear...    542
20  PW CBSE Class 10th PYQs - Past 10 Years' Solve...    499
44  All In One Science CBSE Class 10th Based On La...    447
19  Diary of a Wimpy Kid: Hot Mess (Book 19) (Diar...    439
47  All In One Social Science CBSE Class 10th Base...    425
40           Orbital: Winner of the Booker Prize 2024    399
33  PW Physics Med Easy 2.0 Notes | Notes, Formula...    399
3   My First Library: Boxset of 10 Board Books for...    399
56  NTA UGC NET '24 Paper 1 By Kvs Madaan|Teaching...    390
17  Educart CBSE Science Class 10 Sample Papers 20...    383
50  Zero to One: Notes on startups, or how to buil...    376
54  All In One English Core class 12th Based On La...    374
12     Ikigai: Japanese secret to long and happy life    364
16  Oswaal CBSE Sample Question Papers Class 10 Sc...    350
49  Oswaal CBSE Sample Question Papers Class 10 So...    349
```

```
<ipython-input-10-177cc4d28b0b>:8: UserWarning: Tight layout not applied. The left and right margins cannot be made large enough to accommodate all axes decorations.
  plt.tight_layout()
```



- **Question 2: Show top Rated Books and Authors with respect to the highest customers rating (i.e., based on rating score): Show your result as a data frame as well as a Bar diagram.**

  - ✓ This code identifies and visualizes the top 25 highest-rated books by converting rating column to numeric using pd.to_numeric() and non-convertible values to NaN.
  - ✓ It sorts the dataset by rating in descending order. For ties, books with higher customers_rated are prioritized. It gets the top 25 books with the highest ratings and prints it.
  - ✓ To visualize a horizontal bar chart is created with titles on the y-axis and ratings on the x-axis. It uses invert_yaxis() to display the highest-rated books at the top. The function plt.show() displays the chart.

```
[ ]  # Convert 'rating' to numeric if needed (already numeric in your dataset)
     df['rating'] = pd.to_numeric(df['rating'], errors='coerce')

     # Get the top-rated books (sort by rating, then by customers_rated for ties)
     top_rated_books = df.sort_values(['rating', 'customers_rated'], ascending=[False, False]).head(25)

     # Display the results
     print("Top 25 Rated Books and Authors:")
     print(top_rated_books[['author', 'title', 'rating']])
```

```
# Plot the results
plt.figure(figsize=(12, 8))
plt.barh(top_rated_books['title'], top_rated_books['rating'], color='orange')
plt.xlabel('Rating', fontsize=12)
plt.ylabel('Books', fontsize=12)
plt.title('Top 25 Rated Books', fontsize=16)
plt.gca().invert_yaxis()  # Invert y-axis for better readability
plt.tight_layout()
plt.show()
```

- **Output:**

```
Top 25 Rated Books and Authors:
                          author  \
28                      Om Swami
1                    Ravi Mantri
2                    Subah Saraf
8            PW (Physics Wallah)
15                       Educart
49       Oswaal Editorial Board
33     Dr. Manish Raj (MR. Sir)
19                    Jeff Kinney
4                     James Clear
0                  Morgan Housel
12             Francesc Miralles
41                    Murty Sudha
45        Paramahansa Yogananda
34                  M Laxmikanth
23                  Team Pegasus
9              Harishankar Parsai
38              Wlliam Dalrymple
11       Oswaal Editorial Board
59              Alex Michaelides
3              Wonder House Books
42                  Robin Sharma
50  Peter Masters, Blake,Thiel
29                Dale Carnegie
43                    Franz Kafka
31  Chitra Banerjee Divakaruni

                                                 title  rating
28  Thirteen Months In The Himalayas : Chronicles ...     5.0
1                          Amma Diarylo Konni Pageelu     4.8
2   The Satvic Revolution: 7 Life-Changing Habits ...     4.8
8   PW CBSE | Class 10 | 15 New Pattern Sample Pap...     4.8
15  Educart CBSE Sample Paper Class 10 2024-25 - S...     4.8
49  Oswaal CBSE Sample Question Papers Class 10 So...     4.8
33  PW Physics Med Easy 2.0 Notes | Notes, Formula...     4.7
19  Diary of a Wimpy Kid: Hot Mess (Book 19) (Diar...     4.7
4                                      Atomic Habits     4.6
0                                The The Psychology of Money     4.6
12     Ikigai: Japanese secret to long and happy life     4.6
41  Grandma's Bag of Stories: Collection of 20+ Il...     4.6
45  Autobiography of a Yogi (Complete Original Edi...     4.6
34  Indian Polity for UPSC (English) by M Laxmikan...     4.6
23  1001 Ultimate Brain Booster Activities for 4+ ...     4.6
9                                    Nithalle Ki Diary     4.6
38  The Golden Road : How Ancient India Transforme...     4.6
11  Oswaal CBSE Sample Question Papers Class 10 En...     4.6
59   THE SILENT PATIENT [Paperback] Michaelides, Alex     4.5
3   My First Library: Boxset of 10 Board Books for...     4.5
42  The 5 AM Club – Own Your Morning. Elevate Your...     4.5
50  Zero to One: Notes on startups, or how to buil...     4.5
29  World’s Greatest Books For Personal Growth & W...     4.5
43                                     Metamorphosis     4.5
31  The Palace of Illusions: 10th Anniversary Edit...     4.5
```

```
<ipython-input-12-93b3dc11eae1>:8: UserWarning: Tight layout not applied. The left and right margins cannot be made large enough to accommodate all axes decorations.
  plt.tight_layout()
```

- **Question 3: Show topmost (10/15) Customer Rated Authors and Books (i.e., based on number of customers): Show your result as a data frame as well as a bokeh.palettes, d3**

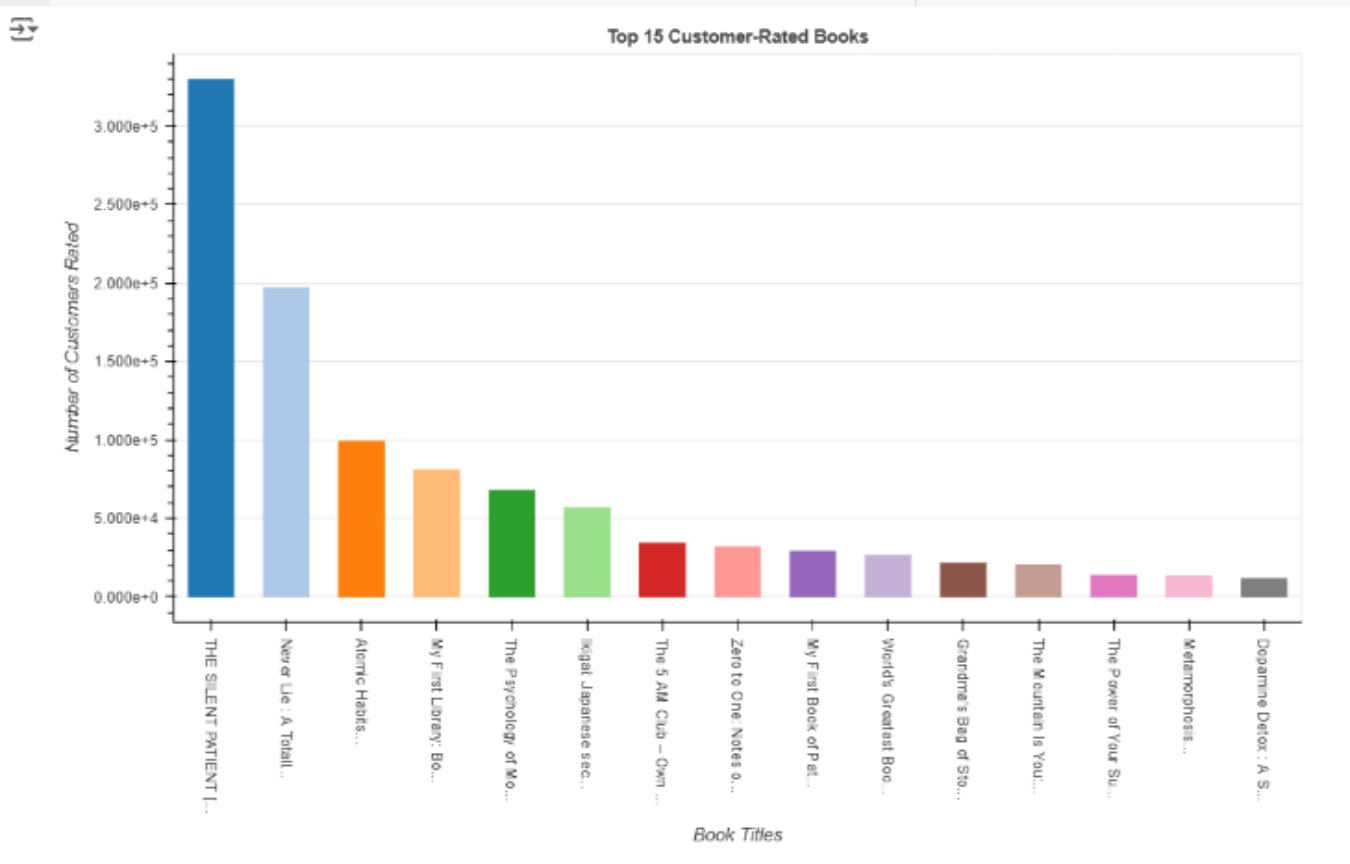
  - ✓ This code identifies and visualizes the top 15 books with the highest customer ratings using the Bokeh library.
  - ✓ It sorts the dataset by customers_rated in descending order and selects the top 15 books. Then we are shortening book titles to the first 20 characters for better readability in the plot.
  - ✓ A bar chart is created with truncated titles on the x-axis and the number of customer ratings (customers_rated) on the y-axis.
  - ✓ We are using the colorful palette “d3['Category20']” for the bars this configures the vertical x-axis labels and custom axis titles for clarity.

```
from bokeh.io import output_notebook, show
from bokeh.plotting import figure
from bokeh.palettes import d3

# Get the top 15 books based on the number of customers who rated them
top_customer_rated = df.sort_values('customers_rated', ascending=False).head(15)

# Display the results
print("Top 15 Customer-Rated Books and Authors:")
print(top_customer_rated[['author', 'title', 'customers_rated']])
```

```
[ ]  from bokeh.io import output_notebook, show
     from bokeh.plotting import figure
     from bokeh.palettes import d3

     # Prepare the data
     top_customer_rated = df.sort_values('customers_rated', ascending=False).head(15)

     # Truncate long titles for better visualization
     top_customer_rated['short_title'] = top_customer_rated['title'].str[:20] + "..."

     # Create the Bokeh plot
     output_notebook()
     p = figure(
         x_range=top_customer_rated['short_title'],  # Use truncated titles
         height=600,
         width=900,
         title="Top 15 Customer-Rated Books",
         toolbar_location=None,
         tools=""
     )

     # Add bars to the plot
     p.vbar(
         x=top_customer_rated['short_title'],
         top=top_customer_rated['customers_rated'],
         width=0.6,
         color=d3['Category20'][15]
     )

     # Adjust x-axis labels
     p.xaxis.major_label_orientation = "vertical"  # Use vertical labels for clarity
     p.xgrid.grid_line_color = None
     p.yaxis.axis_label = "Number of Customers Rated"
     p.xaxis.axis_label = "Book Titles"
     p.title.align = "center"

     # Show the plot
     show(p)
```

- **Output:**

```
Top 15 Customer-Rated Books and Authors:
                          author  \
59              Alex Michaelides
32  The Housemaid Freida McFadden
4                   James Clear
3             Wonder House Books
0                  Morgan Housel
12              Francesc Miralles
42                  Robin Sharma
50    Peter Masters, Blake,Thiel
13            Wonder House Books
29                 Dale Carnegie
41                   Murty Sudha
18                 Brianna Wiest
7                  Joseph Murphy
43                   Franz Kafka
21              Thibaut Meurisse

                                          title  customers_rated
59   THE SILENT PATIENT [Paperback] Michaelides, Alex          329614
32  Never Lie : A Totally Gripping Thriller with M...          196948
4                                     Atomic Habits            98774
3   My First Library: Boxset of 10 Board Books for...            80744
0                           The Psychology of Money            67648
12     Ikigai: Japanese secret to long and happy life            56499
42  The 5 AM Club – Own Your Morning. Elevate Your...            33987
50  Zero to One: Notes on startups, or how to buil...            31610
13  My First Book of Patterns Pencil Control: Patt...            28924
29  World's Greatest Books For Personal Growth & W...            26276
41  Grandma's Bag of Stories: Collection of 20+ Il...            21254
18  The Mountain Is You: Transforming Self-Sabotag...            20103
7   The Power of Your Subconscious Mind: Original ...            13452
43                                    Metamorphosis            13169
21  Dopamine Detox : A Short Guide to Remove Distr...            11285
```



Step:6 Comparing the performance of other Web Scraping Tools.

Question 4: Complete the above tasks using at least three separate web craping packages and compare their performance (processing time).

- **Scraping Functions:**
  - ✓ bs_test: Uses BeautifulSoup to parse the HTML and extract book titles.
  - ✓ lxml_test: Uses lxml's XPath to locate and extract the book titles.
  - ✓ regex_test: Uses regular expressions (regex) to find book titles by matching patterns in the HTML.

```python
import re
import time
import requests
from bs4 import BeautifulSoup
from lxml import html as lxmlhtml

# Function to scrape using BeautifulSoup
def bs_test(html):
    soup = BeautifulSoup(html, 'html.parser')
    book_elements = soup.find_all('div', class_='_cDEzb_iveVideoWrapper_JJ34T')
    books = []
    for book_element in book_elements:
        title = book_element.find('div', class_='_cDEzb_p13n-sc-css-line-clamp-1_1Fn1y')
        title = title.get_text(strip=True) if title else "Unknown Title"
        books.append({'title': title})
    return books

# Function to scrape using lxml
def lxml_test(html):
    tree = lxmlhtml.fromstring(html)
    book_elements = tree.xpath("//div[contains(@class, '_cDEzb_iveVideoWrapper_JJ34T')]")
    books = []
    for book in book_elements:
        title = book.xpath(".//div[contains(@class, '_cDEzb_p13n-sc-css-line-clamp-1_1Fn1y')]/text()")
        title = title[0].strip() if title else "Unknown Title"
        books.append({'title': title})
    return books

# Function to scrape using regex
def regex_test(html):
    books = []
    matches = re.findall(r'<div.*?class=".*?_cDEzb_p13n-sc-css-line-clamp-1_1Fn1y.*?">(.*?)</div>', html)
    for title in matches:
        books.append({'title': title.strip()})
    return books
```

```python
# Function to measure execution time and count books
def timeit_and_count(fn, html):
    t1 = time.time()
    books = None
    for _ in range(10):  # Repeat 10 times for a fair comparison
        books = fn(html)
    t2 = time.time()
    print(f'{fn.__name__} found {len(books)} books and took {(t2 - t1) * 1000:.3f} ms')

# Main function to fetch HTML and perform tests
if __name__ == "__main__":
    # Base URL and headers
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36'
    }
    base_url = "https://www.amazon.in/gp/bestsellers/books"

    # Fetch HTML for the first page
    response = requests.get(base_url, headers=headers)
    html = response.text

    # Measure performance and count books
    for fn in (bs_test, lxml_test, regex_test):
        timeit_and_count(fn, html)
```

```
bs_test found 30 books and took 1052.614 ms
lxml_test found 30 books and took 136.316 ms
regex_test found 58 books and took 1299.619 ms
```

- **Code Explanation:**

  - ✓ This code compares the performance of three HTML scraping methods (BeautifulSoup, lxml, and regex) in extracting book titles from a webpage.
  - ✓ timeit_and_count function measures the execution time and counts the number of books extracted by each function. Each function is executed 10 times to get a fair comparison of performance.
  - ✓ It fetches the HTML content of the Amazon bestseller books page using a custom user-agent header. Iterates over the three scraping functions, calling timeit_and_count to compare how quickly they can scrape book titles.
  - ✓ Regex is consuming the maximum time comparing beautifulsoup and Lxml.

Conclusion

Objectives Achieved:

We are able to successfully scrape the book details, and we saved them in a structured format. The dataset we got is cleaned and preprocessed for analysis. We finally derived insights into top-rated and highest-priced books from the dataset. The visualizations provides a clear insights into the data.

Challenges Faced:

Due to the issues with the dynamic webpage structures, we are able to scrape only 60 books from a webpage of 100 books. Handling missing data and accurate parsing requires specific and complex logic.

****