HARNESSING WEB SCRAPING TO ANALYZE AMAZON PRODUCT TRENDS

TERM PROJECT BY:

ADHIMULAM BHARGAV SAI VISWANTH

REGD.NO: 23BQ1A4201

PROJECT GUIDE :

Mr. M. PARDHA SARADHI,

Associate Professor

ABSTRACT

- The project involves web scraping Amazon product data for market research and competitive analysis.
- Using libraries like BeautifulSoup, Selenium, Scrapy, and Pandas, we will extract key product information (e.g., name, price, ratings, and offers) from Amazon URLs.
- The data will be stored in structured formats (CSV/Excel) for analysis, enabling insights into market trends, pricing strategies, and product performance.
- The final output will include cleaned datasets, visualizations, and a comprehensive report.

INTRODUCTION

- The rise of e-commerce has created vast amounts of valuable data.
- Web scraping offers a solution to extract and analyze this data.
- Amazon, as a leading e-commerce platform, provides valuable insights.
- This project focuses on extracting product-specific data from Amazon.
- The extracted data will aid in understanding market trends and product performance.
- Web scraping faces challenges, including anti-scraping measures.
- Python libraries (BeautifulSoup, Selenium, Scrapy, Pandas) will facilitate data extraction.
- The project's outcome will inform business decisions and market research.
- A comprehensive report will detail the project findings.
- The project's success relies on efficient data extraction and analysis.
- The extracted data will provide valuable insights into competitive analysis.
- This project contributes to the field of market research and analysis.

PROBLEM STATEMENT

Challenges:

- **Manual Collection**: Extracting product data (titles, reviews, delivery options) from e-commerce sites like Amazon is time-consuming and prone to errors.
- **Scalability Issue**: Navigating multiple pages and handling large datasets manually is inefficient and unscalable.
- **Unstructured Data**: Product information is embedded in unstructured HTML, requiring advanced parsing techniques for meaningful insights.

Solution:

- **Web Scraping with Python**: Automates data extraction, ensuring accuracy and scalability for large datasets.
- **Data-Driven Insights**: Enables efficient analysis of reviews, ratings, and delivery trends to support informed decision-making.

Sample Output

1. Scraped Data Output:

- The scraped product data (titles, reviews, ratings, delivery information, etc.) is saved in a structured format such as a **CSV file**.
- **Example:** Display a small table snippet of your CSV data. For instance:

S.NO	PRODUCT NAME	PRODUCT ID	PRICE	RATING	NUMBER OF REVIEWS	DELIVERY DATE	PRODCUT DESCRIPTION
1	Samsung Galaxy M15 5	B0DGXBSYHS	11999	3.9	832	Tue, 26 Nov	Displaying Description

2. Data Utilization for Visualizations:

- The saved data is used to create **visualizations** such as: Line Chart, Bar Chart, Pie Chart.

3. Visualization Examples:

1. A line chart showing product names on the x-axis and the number of reviews on the y-axis.
2. A bar chart depicting Price distribution of each product.

Tools and Libraries

"Python: The core programming language used."

"BeautifulSoup: For HTML parsing and extracting data."

"Pandas: For data manipulation and cleaning."

"Matplotlib: For creating visualizations."

"Seaborn: For advanced statistical plots."

“Sends HTTP requests to web pages and retrieves their content.”

Data Selection

For the project, product data was scraped from specific categories on Amazon that are relevant to both students and parents:

1. **Electronics**: Smartphones
2. **Cosmetics**: Moisturizing Face Cream
3. **Health & Beauty**: Organic Shampoos for Dry Hair
4. **Computers & Accessories**: Laptops
5. **Home & Kitchen**: Non-Stick Pans

Criteria for Selection

- **Relevance**: Popular and commonly purchased product types.
- **Diversity**: Categories were chosen to represent a variety of needs and interests.
- **Feasibility**: Products with enough reviews and structured data for meaningful analysis.

Data Points Collected

For each product, the following details were extracted:

Product Name, Product ID, Price, Ratings, Number Of Reviews, Delivery Information, Product Description.

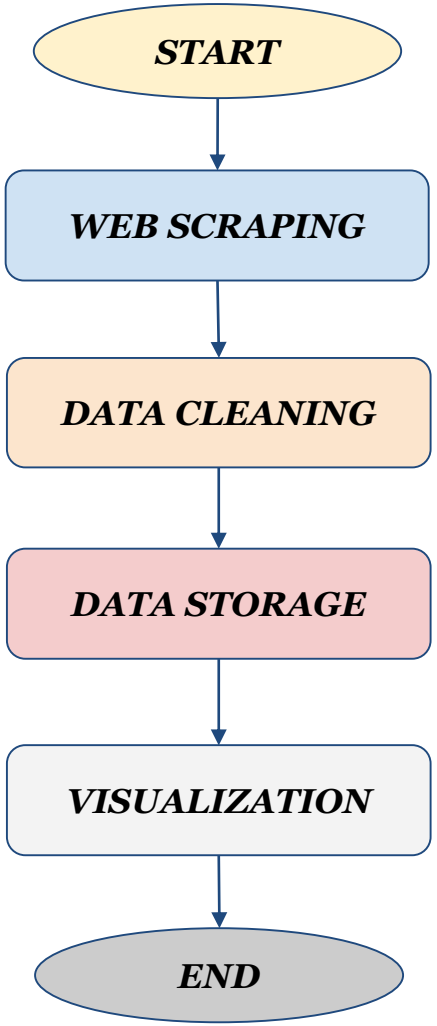
Examples of URLs Used

- Example 1: "https://www.amazon.in/s?k=smartphones&ref=nb_sb_noss"
- Example 2: “https://www.amazon.in/s?k=moisturizing+face+cream&ref=nb_sb_noss”

Methodology

Key Steps in the Process

- **Web Scraping**
  - **Description**:
    Used **BeautifulSoup** to extract product data (e.g., product names, prices, ratings, reviews, and delivery information) from the HTML content of Amazon pages.
- **Data Cleaning**
  - **Description**:
    Ensured the data was accurate and ready for analysis by performing:
- **Data Storage**
  - **Description**:
    Stored the cleaned data in a **CSV file** for easy access and further processing.
- **Data Visualization**
  - **Description**:
    Created charts and graphs to analyze and present the data.



Code Snippets

Some of the logical components, such as:

- **Web Scraping:** Showing how data is fetched from the website such as product title.

```
# Extracting Product title
products = soup.find_all('div', {'data-component-type': 's-
search-result'})
for product in products:
raw_title = product.h2.text.strip() if product.h2 else "-"
print("Product:", raw_title)
-------------------------------------------------------------
#Sample Output:
Product: Samsung Galaxy S21 5G
Product: Apple iPhone 14 Pro Max
Product: OnePlus 11R 5G
Product: Redmi Note 12 Pro
Product: Realme Narzo 60
```

- Data Cleaning: Highlight important cleaning techniques.

```
Data Cleaning Output
data['Price'] = data['Price').replace("-", 0).astype(float)
data['Rating'] = data['Rating').fillna(0)
data['Number of Reviews'] = data['Number of
Reviews'].fillna(0).astype(int)
median_price = data.loc[data['Price'] != 0, 'Price'].median()
data['Price'] = data['Price').replace(0, median_price)
print(data.head())
```

Output:Before Data Cleaning

Product Name	Price	Rating	Number of Reviews
iPhone 13	999.99	4.5	1000
Samsung Galaxy S21	-	NaN	800
OnePlus 9	549.99	4.7	NaN
Redmi Note 10	200	4	500
Realme X7	-	4.2	NaN

Output:After Data Cleaning

Product Name	Price	Rating	Number of Reviews
iPhone 13	999.99	4.5	1000
Samsung Galaxy S21	549.99	0	800
OnePlus 9	549.99	4.7	0
Redmi Note 10	200	4	500
Realme X7	549.99	4.2	0

Data Storage: Show how data is saved (e.g., to a CSV).

```
#Data Stored in a csv file
data.to_csv('amazon_products.csv', index=False)
```

Sample Output

S.NO	PRODUCT NAME	PRODUCT ID	PRICE	RATING	NUMBER OF REVIEWS	DELIVERY DATE	PRODCUT DESCRIPTION
1	Samsung Galaxy M15 5	B0DGXBSYHS	11999	3.9	832	Tue, 26 Nov	Displaying Description

VISUALIZATION ANALYSIS (STATISTICS)

Here is the Statistical Visualisation Analysis of Sample Scrapped Data:-

- **Visualization 1**: Scraped Data (Table Format)
- **Visualization 2**: Price Distribution of Each Product (Bar Chart)
- **Visualization 3**: Product Name vs. Number of Reviews (Line Chart)
- **Visualization 4**: Rating Distribution Across Products (Pie Chart)
- **Visualization 5**: Top Most Expensive Products Based on Price (Horizontal Bar Chart)
- **Visualization 6**: Bottom Least Expensive Products Based on Price (Horizontal Bar Chart)

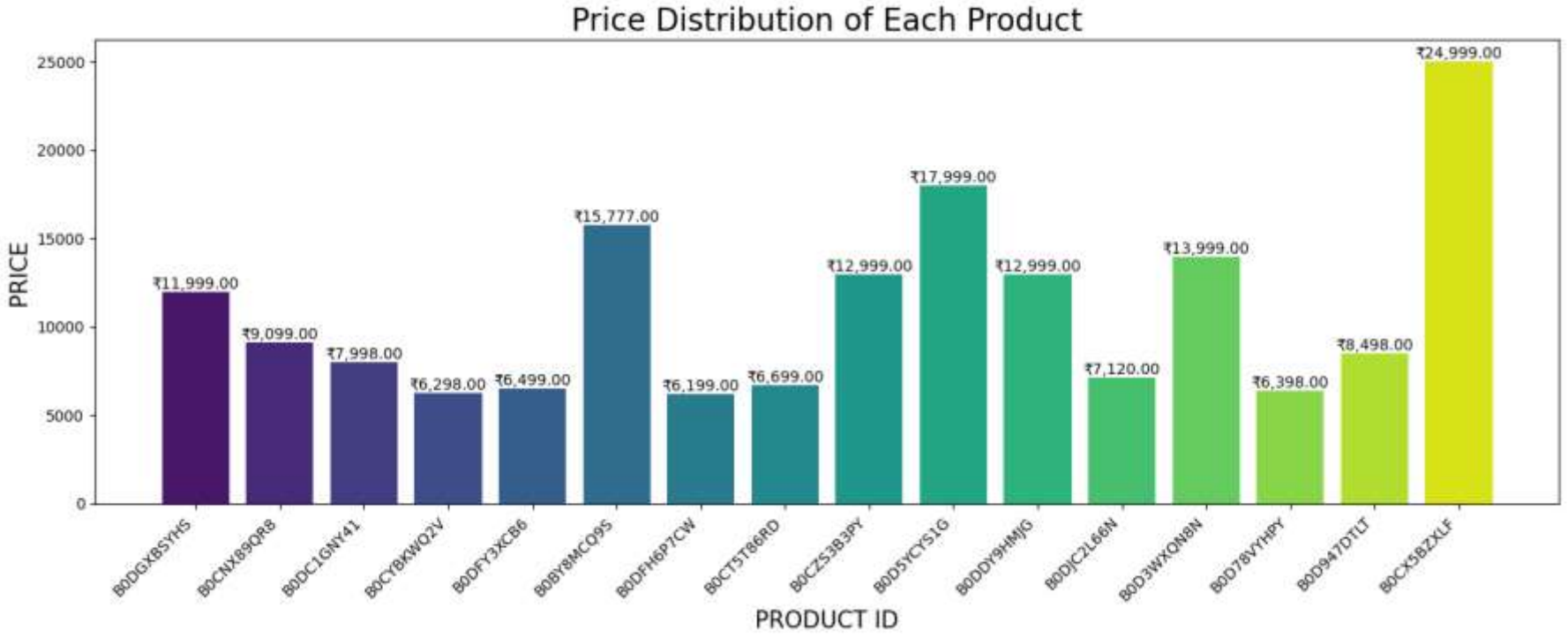
- **Visualization 1**: Scraped Data (Table Format)

Provides a summary of key details such as product names, prices, ratings, reviews, and delivery information.

	S.NO	PRODUCT NAME	PRODUCT ID	PRICE	RATING	NUMBER OF REVIEWS	DELIVERY DATE	PRODUCT DESCRIPTION
0	1	Samsung Galaxy M15 5	B0DGXBSYHS	11999.00	3.9	832	Tue, 26 Nov	Samsung Galaxy M15 5G Prime Edition (Stone Grey,6GB RAM,128GB Storage)| Super AMOLED Display| 50MP Triple Cam| 6000mAh Battery| MediaTek Dimensity 6100+| 4 Gen. OS Upgrade & 5 Year Security Update
1	2	Redmi 13C 5G	B0CNX89QR8	9099.00	3.9	97	Tue, 26 Nov	Redmi 13C 5G (Starlight Black, 4GB RAM, 128GB Storage)	MediaTek Dimensity 6100+ 5G	90Hz Display
2	3	POCO M6 5G, Orion Bl	B0DC1GNY41	7998.00	3.9	58	Tue, 26 Nov	POCO M6 5G, Orion Blue (4GB, 64GB)
3	4	POCO C61 Mystical Gr	B0CYBKWQ2V	6298.00	3.4	36	Tue, 26 Nov	POCO C61 Mystical Green 4GB RAM 64GB ROM
4	5	Samsung Galaxy M05	B0DFY3XCB6	6499.00	3.9	870	Delivery info not available	Samsung Galaxy M05 (Mint Green, 4GB RAM, 64 GB Storage)	50MP Dual Camera	Bigger 6.7" HD+ Display	5000mAh Battery	25W Fast Charging	2 Gen OS Upgrade & 4 Year Security Update	Without Charger
5	6	OnePlus Nord CE 3 Li	B0BY8MCQ9S	15777.00	4.2	97	Tue, 26 Nov	OnePlus Nord CE 3 Lite 5G (Chromatic Gray, 8GB RAM, 128GB Storage)
6	7	Lava O3	B0DFH6P7CW	6199.00	3.8	42	Tue, 26 Nov	Lava O3 (Glossy Black, 4 GB RAM, 64 GB Storage)	Biggest 6.75" HD+ Display	13MP AI Dual Rear Camera	5000 mAh Battery	Secure Face Unlock	Fingerprint Reader	Charger & Phone-Cover in Box
7	8	Lava Yuva 3	B0CT5T86RD	6699.00	4.0	299	Thu, 28 Nov	Lava Yuva 3 (Cosmic Lavender, 4+4*GB + 128GB)	Segment's Most Affordable Smartphone with 128 GB (UFS 2.2) Storage	90Hz Punch Hole Display	13MP AI Triple Camera	Side Fingerprint Sensor	Bloatware Free
8	9	realme NARZO 70x 5G	B0CZS3B3PY	12999.00	4.0	97	Tue, 26 Nov	realme NARZO 70x 5G (Ice Blue, 6GB RAM,128GB Storage)|120Hz Ultra Smooth Display|Dimensity 6100+ 6nm 5G|50MP AI Camera|45W Charger in The Box
9	10	OnePlus Nord CE4 Lit	B0D5YCYS1G	17999.00	4.1	97	Tue, 26 Nov	OnePlus Nord CE4 Lite 5G (Super Silver, 8GB RAM, 128GB Storage)
10	11	Motorola G45 5G	B0DDY9HMJG	12999.00	3.8	44	Thu, 28 Nov	Motorola G45 5G (Brilliant Blue, 8GB RAM, 128GB Storage)
11	12	Samsung Galaxy F05	B0DJC2L66N	7120.00	3.0	2	Tue, 26 Nov	Samsung Galaxy F05 (Twilight Blue, 64 GB) (4 GB RAM)
12	13	realme NARZO 70x 5G	B0D3WXQN8N	13999.00	4.0	97	Tue, 26 Nov	realme NARZO 70x 5G (Ice Blue, 8GB RAM,128GB Storage)|120Hz Ultra Smooth Display|Dimensity 6100+ 6nm 5G|50MP AI Camera|45W Charger in The Box
13	14	Redmi A3X	B0D78VYHPY	6398.00	3.7	136	Tue, 26 Nov	Redmi A3X (Midnight Black, 3GB RAM, 64GB Storage)	Premium Halo Design	90Hz Display	Powerful Octa Core Processor
14	15	realme NARZO N61	B0D947DTLT	8498.00	4.0	988	Thu, 28 Nov	realme NARZO N61 (Voyage Blue,6GB RAM+128GB Storage) 90Hz Eye Comfort Display	IP54 Dust & Water Resistance	48-Month Fluency	Charger in The Box
15	16	Oneplus Nord CE4	B0CX5BZXLF	24999.00	4.2	97	Tue, 26 Nov	Oneplus Nord CE4 (Dark Chrome, 8GB RAM, 256GB Storage)

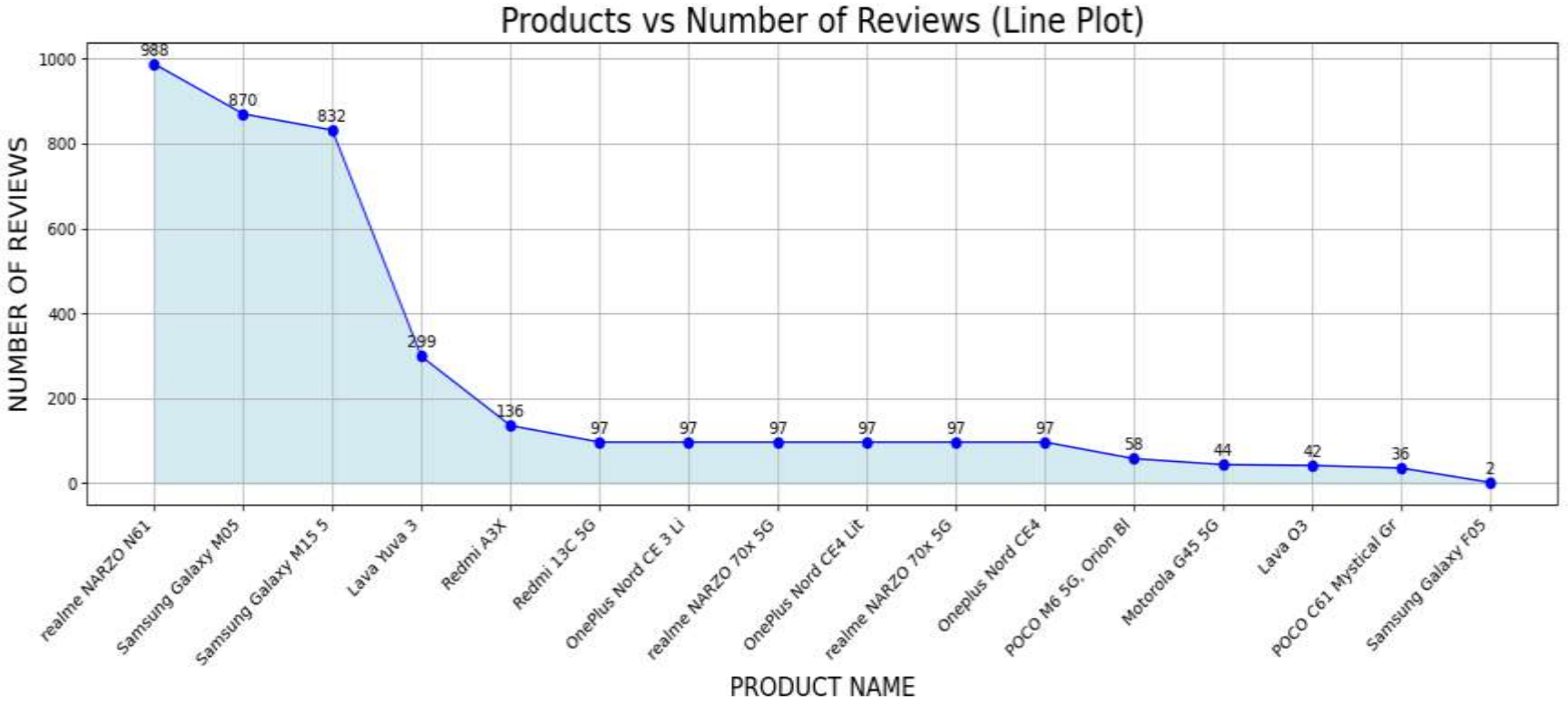
- **Visualization 2**: Price Distribution of Each Product (Bar Chart)

This bar chart illustrates the distribution of product prices within the dataset, with individual prices displayed alongside their respective product IDs. It provides an overview of the pricing spectrum for the selected products.



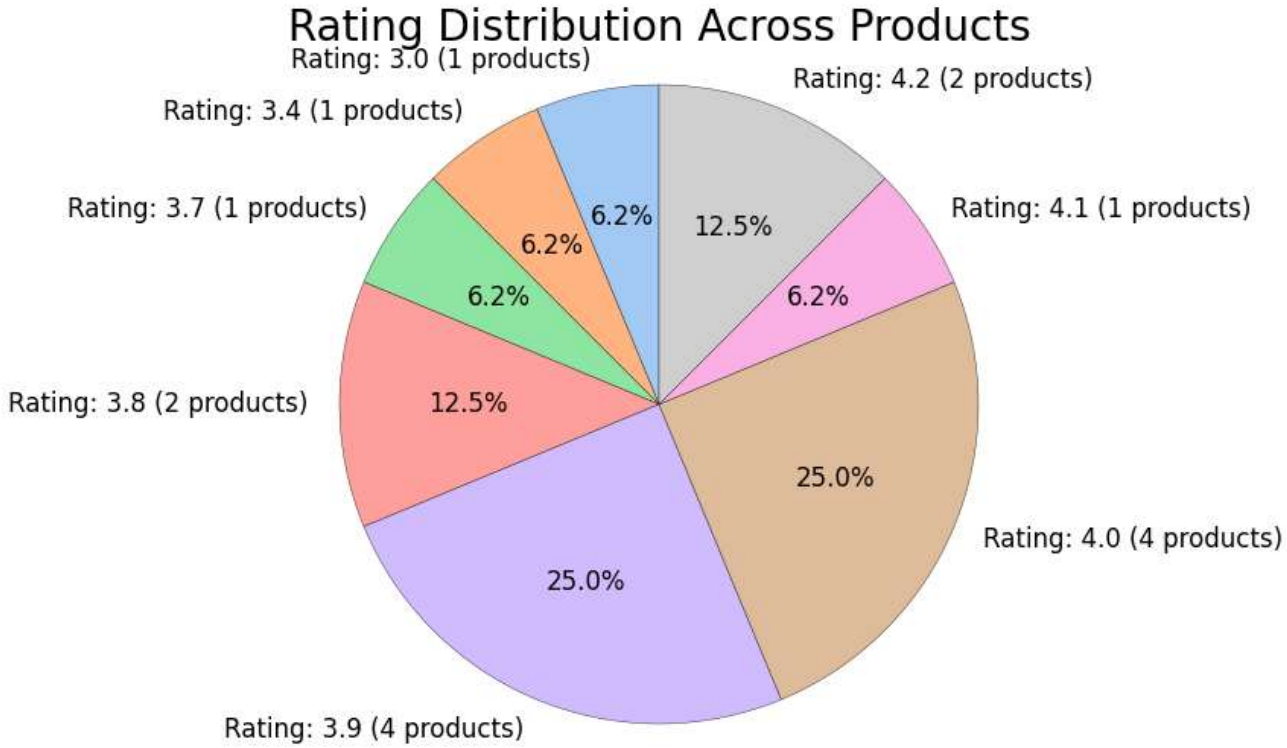
- **Visualization 3**: Product Name vs. Number of Reviews (Line Chart)

This line plot displays the number of reviews for various products, reflecting consumer feedback and popularity. Each point on the line corresponds to a product name and the total number of reviews it received



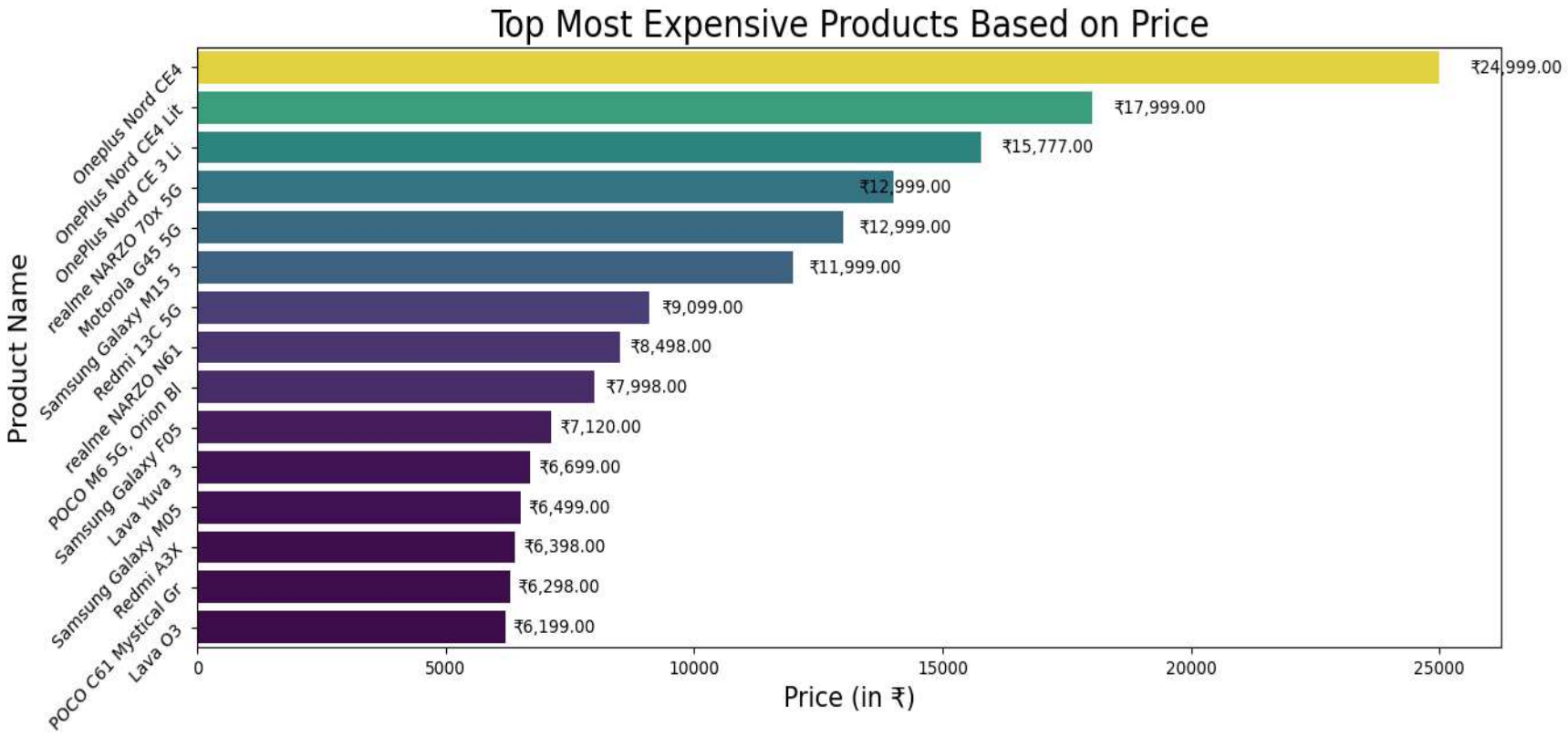
- **Visualization 4**: Rating Distribution Across Products (Pie Chart)

This pie chart illustrates the distribution of ratings across the dataset, indicating the percentage of products with specific ratings. The chart highlights customer satisfaction trends and rating frequencies.



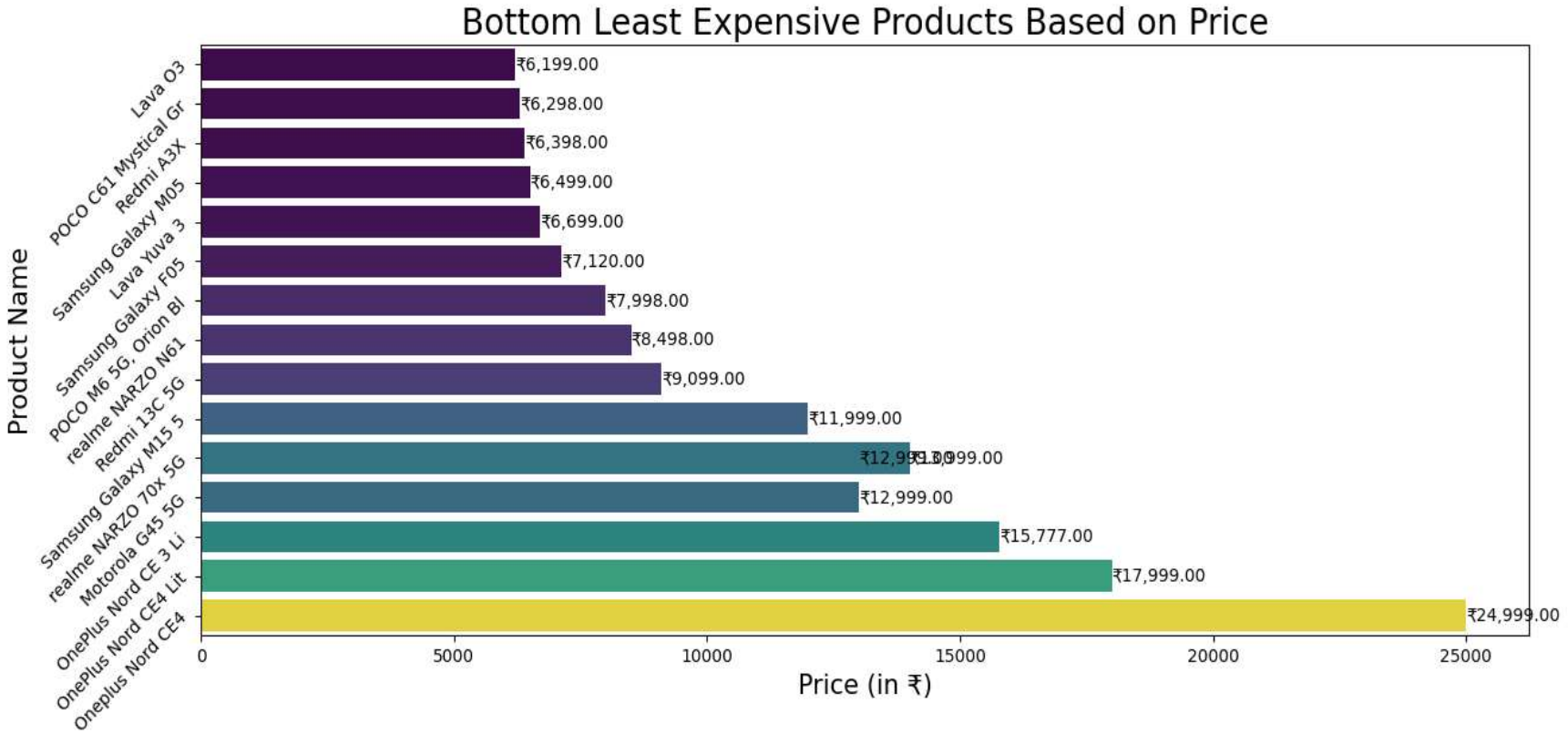
- **Visualization 5**: Top Most Expensive Products Based on Price (Horizontal Bar Chart)

This bar chart highlights the most expensive products in the dataset, ranked by their prices. The chart provides a clear comparison of premium-priced items.



- **Visualization 6**: Bottom Least Expensive Products Based on Price (Horizontal Bar Chart)

This bar chart highlights the least expensive products in the dataset. It provides a clear view of pricing trends among the most affordable products, with their prices plotted alongside the



Challenges and Solutions

Challenges	Solutions
Handling Inconsistent Data: Missing values, inconsistent formatting	Used data cleaning techniques to fill missing values (e.g., replacing NaN with 0 for ratings) and standardized the data format
Web Scraping Restrictions: Getting blocked or rate-limited by Amazon	Implemented exponential backoff with jitter and used browser headers to avoid detection
Parsing Complex HTML Structure: Difficulties extracting data from diverse page structures	Used BeautifulSoup and targeted specific CSS selectors and classes to extract relevant product details
Storing Large Datasets: Handling large amounts of scraped data	Saved data into CSV files and processed them efficiently using Pandas DataFrame
Visualizing Data Effectively: Presenting large datasets clearly	Created simple, readable charts using Matplotlib and Seaborn to show trends in product prices, ratings, and reviews
Managing Errors and Timeouts: HTTP errors during scraping	Used retry mechanisms with exponential backoff to handle temporary network issues
Data Overload: Overwhelming amounts of data when scraping multiple categories	Limited scraping to fewer pages and stored data incrementally to avoid overloading memory

Conclusion

1. **Automated Web Scraping**: Used Python libraries (BeautifulSoup, Pandas) to efficiently extract and clean product data from Amazon.
2. **Error Handling**: Implemented exponential backoff and retry mechanisms for robustness against errors and restrictions.
3. **Data Storage**: Stored scraped data in CSV files for easy handling of large datasets.
4. **Data Visualization**: Used Matplotlib and Seaborn to visualize trends in product pricing, ratings, and reviews.
5. **Challenges Addressed**: Tackled inconsistent data, scraping restrictions, and large datasets through cleaning, retry logic, and optimized storage.
6. **Outcome**: Demonstrated the efficiency of automation for large-scale data collection and analysis.

THANK YOU