Amazon Web Scraper Project Report

1. Introduction Web scraping is a powerful technique for extracting data from websites. This project involves developing an Amazon Web Scraper to collect product information, including titles, prices, and links. The extracted data is saved in a CSV file for further analysis. The project leverages Python libraries such as `requests` and `BeautifulSoup` to interact with Amazon's website.

2. Objectives

- To automate the extraction of product details from Amazon.
- To store scraped data in a structured format (CSV).
- To analyze the extracted product data using Python.

3. Technologies Used

- **Programming Language:** Python
- **Libraries:**
  - `requests` (to send HTTP requests)
  - `BeautifulSoup` (to parse HTML content)
  - `pandas` (to handle data processing)
  - `csv` (to store data in CSV format)
  - `time` and `random` (to add delays to mimic human behavior)

4. Methodology

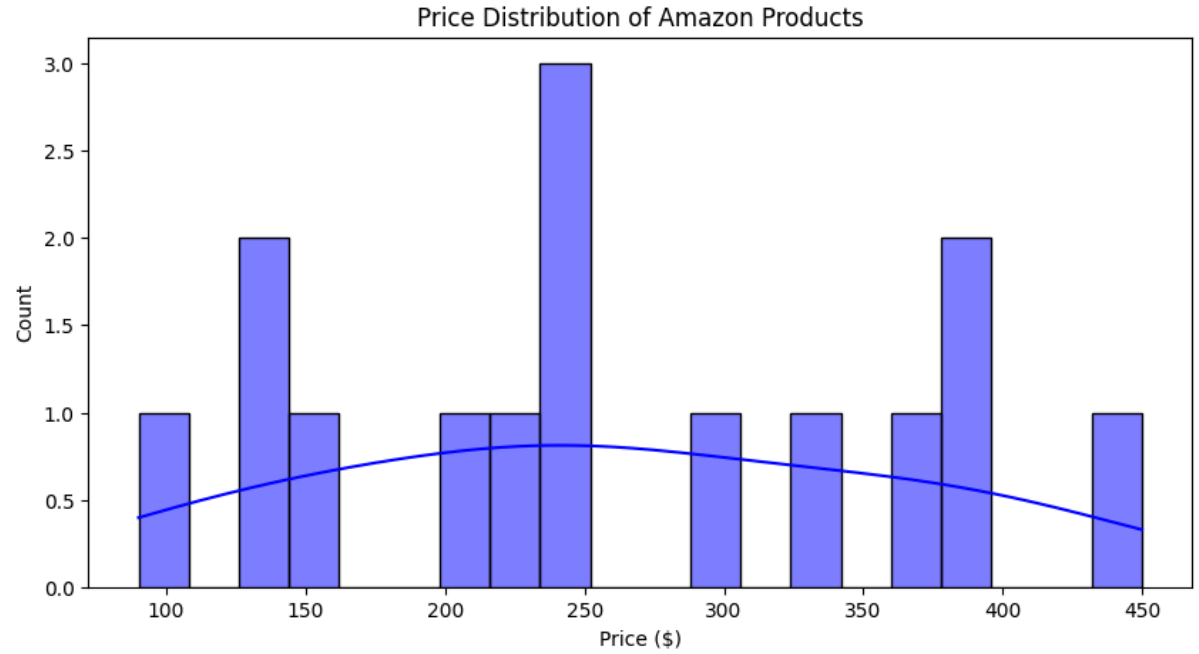
- **Data Extraction:**
  - Send an HTTP request to Amazon's search results page.
  - Parse the HTML response to extract product details.
  - Extract relevant details such as product title, price, and link.
- **Data Storage:**
  - The extracted data is structured and saved into a CSV file.
- **Data Analysis:**
  - Load the data into a Pandas DataFrame.
  - Display basic statistics and visualization.

5. Implementation

- **Scraping Function:**
  - Sends requests to Amazon's search results page using a predefined user-agent.
  - Parses the HTML using `BeautifulSoup`.
  - Extracts product title, price, and URL.
  - Handles missing or unavailable data.
- **Saving to CSV:**
  - Uses `csv.DictWriter` to save data in a structured format.
- **Execution:**

- The script takes a search term as input.
- Scrapes product details and saves them to `amazon_products.csv`.
- Outputs the number of products scraped.

Visualisations



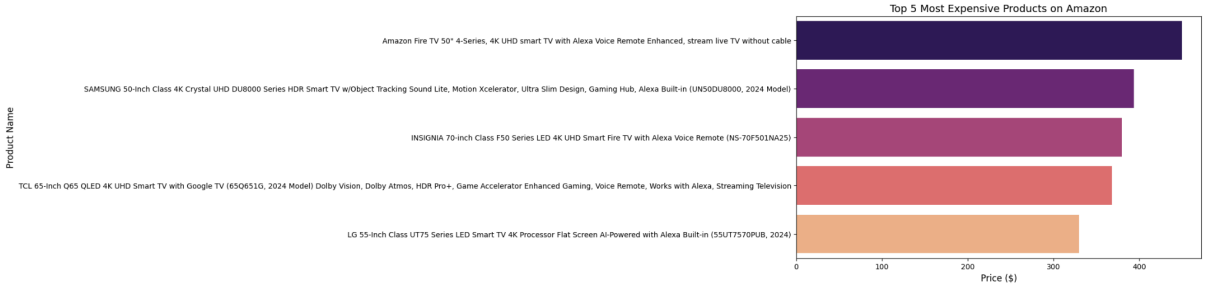


Chart Description:

Title: Price Distribution of Amazon Products

X-axis: Price in dollars ($100 to $450)

Y-axis: Count (0.0 to 3.0)

Bars: Represent the frequency of products within specific price ranges.

Trend Line: A blue curve overlaid on the bars, indicating the trend of product prices

Key Observations:

Peak Frequency:

The highest frequency of products is observed at the $250 price point, with a count of 3.0. This indicates that $250 is the most common price for Amazon products in this dataset.

Other Notable Price Points:

The $150 and $400 price points each have a count of 2.0, suggesting that these price ranges are also relatively popular among Amazon products.

Trend Line Analysis:

The blue trend line peaks around the $250 price range, showing that the majority of products are clustered around this price. The trend gradually decreases towards both higher and lower price ranges, indicating fewer products in those areas.

Insights and Implications:

Market Strategy:

Knowing that $250 is a common price point, sellers might consider pricing their products around this range to tap into the most frequent price bracket.

Consumer Preferences:

The distribution suggests that consumers are more likely to purchase products priced around $250. Understanding this can help in targeting marketing efforts and promotions.

Price Elasticity:

Products at $150 and $400 also show significant frequency, indicating that consumers are willing to spend at these price points as well. This can be useful for pricing strategies and discount offers.

6. Challenges and Solutions

- **Amazon's anti-scraping measures:**
  - Used headers with a user-agent to mimic a real browser.
  - Introduced random delays to reduce detection.
- **Incomplete data extraction:**
  - Implemented exception handling to skip missing values.
- **Dynamic page structures:**
  - Used multiple CSS selectors to adapt to variations.

7. Results

- Successfully scraped product details from Amazon.
- Stored data in `amazon_products.csv`.
- Loaded data into Pandas for further analysis.