1. Introduction

The name of this project is Online Book Price and Rating Analysis. I completed this project individually. The main idea is to look at how book prices and book ratings behave on an online platform and to see whether there is any obvious relationship between them.

Originally, I planned to scrape product information from Sephora or Ulta Beauty, because I wanted to analyze beauty product pricing trends. However, those websites blocked most of my requests. I tried using custom headers, user-agent strings, cookies, and different request intervals, but I still received repeated 403 errors. Their API access also required tokens or paid usage, so it was not realistic for this course project. Because of that, I switched to Books to Scrape, a public demo website created for practicing web scraping. The site structure is simple, predictable, and does not block crawlers, which allowed me to complete the full pipeline of scraping, data cleaning, analysis, and visualization. Although the dataset is small, it still provides a clear way to practice the techniques taught in class.

This project uses Python to collect and process book information and then analyzes how price, rating, and high-priced books appear in the dataset.

2. Data Collection

The dataset comes from the website Books to Scrape. I scraped the first five pages of the catalogue section. Each page contains 20 books, so the final dataset has 100 book entries.

Scraping Process

I used the requests library to send HTTP GET requests and BeautifulSoup to parse the HTML. Each book is inside an <article class="product_pod"> element, so I extracted: book title (<h3>), price (<p class="price_color">), rating (stored as a CSS class like “star-rating Three”), URL of the book’s detail page (relative link).

A small technical detail is that the rating is not given as a number. I mapped the textual ratings (“One”, “Two”, “Three”, “Four”, “Five”) to numerical scores 1–5 so that the data could be analyzed more easily.

For pagination, I looped through page numbers 1 to 5 and dynamically constructed URLs such as: http://books.toscrape.com/catalogue/page-1.html

This allowed the scraper to collect multiple pages automatically instead of manually downloading any files.

Data Cleaning

After scraping, I cleaned the following:

- Price: removed the "£" symbol and converted values to floats
- Rating: converted text rating into integers
- Duplicates: checked for repeated titles (none in this dataset)
- Missing values: ensured each field existed before saving

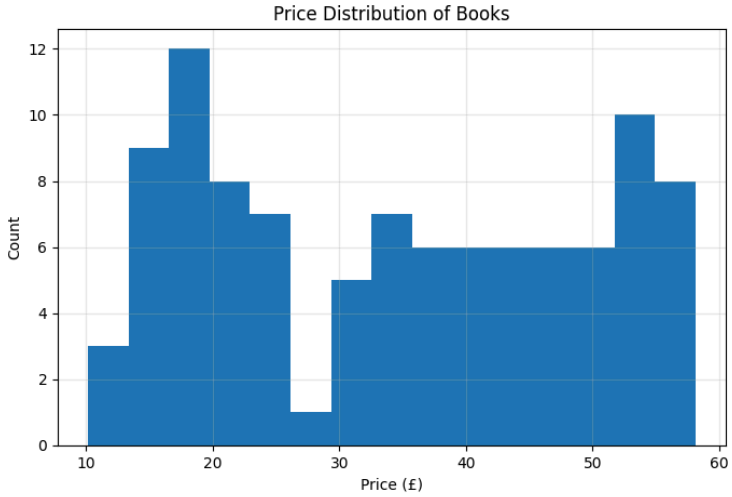
The cleaned data was stored in a processed CSV file in the project's data/processed folder, following the project's required structure.

The biggest change from my original proposal was switching topics entirely because of website restrictions. The scraping itself became simple after changing the source, but it still covered the full workflow of web data collection taught in class.

3. Analysis and Visualizations

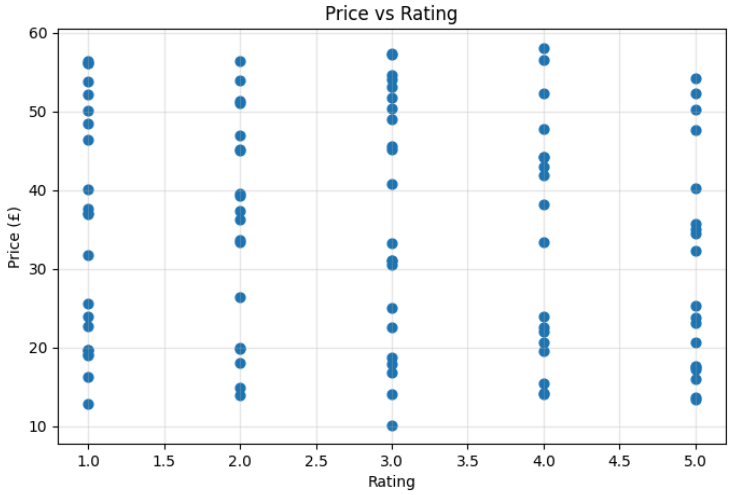
Because the dataset only includes title, price, rating, and URLs, I focused on simple descriptive statistics and visualizations using Matplotlib.

Below are the main results and interpretations.

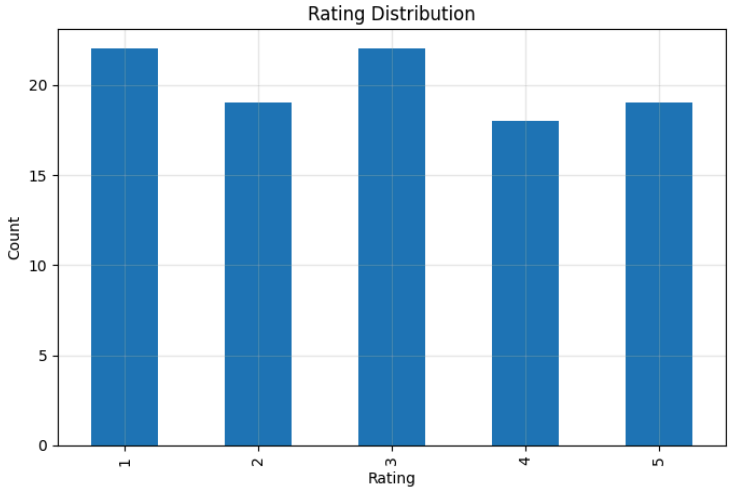


The histogram shows how book prices are spread across the dataset. Most books fall between £15 and £40, and this middle range forms the tallest bars. Only a few books go above £50. This suggests the website uses a centered pricing strategy, with relatively stable prices and very few

extreme values. I used default bin settings from Matplotlib, which already grouped the values clearly.



In this plot, the rating is on the x-axis (1–5), and price is on the y-axis. Each point represents one book. The points do not form any upward or downward trend. Books with the same rating can be very cheap or quite expensive. Even books rated 4 or 5 stars still show a wide range of prices. This means rating does not seem to affect price, and price also does not predict rating. This is different from what I expected at first — I thought higher-rated books might cost more, but the data does not show that.



The bar chart for ratings shows that each rating level (from 1 to 5) appears in similar quantities. There is no strong bias toward high-rated or low-rated books. This balanced distribution keeps the dataset stable but also limits the amount of variation we can analyze. For example, it becomes harder to compare “high-rated vs. low-rated” books because the groups are almost the same size.

This horizontal bar chart highlights the ten highest-priced books in the scraped pages.

Most of these titles:

- have longer names
- look like special-topic or academic-style books
- are far more expensive than regular novels or simple titles

This suggests that the high-price group mostly consists of niche categories, not mainstream popular books.

4. Observation and Conclusion

From these figures and the cleaned dataset, several patterns appear: Prices mostly stay in the middle range, indicating a steady pricing style instead of drastic variation. Ratings are evenly distributed, so the site does not emphasize heavily rated products. Price and rating have no strong relationship—books with the same rating can have very different prices, and expensive books appear across multiple rating levels. The highest prices mainly come from special-topic books, not typical fiction titles.

Overall, the website's pricing seems more connected to book type rather than user rating. A pricey book is not necessarily rated higher, and a highly rated book is not necessarily more expensive.

5. Impact of Findings

Although the dataset is small, the results give a basic idea of how this particular platform lists books. One takeaway is that consumers should not assume that price reflects rating or popularity, at least not on this website.

For sellers or platform designers, the pattern shows that the pricing structure stays within a few common ranges, while only certain categories reach the highest prices. This may help with decisions about promotions or category labeling.

Because the dataset only includes simple information, the conclusions are not meant to reflect the entire online book market, but they still provide an initial look at how price and rating behave separately.

6. Future Work

If more time were available, the project could be extended in several ways. One simple improvement would be to scrape more pages or even look at additional book websites, so the price patterns could be compared across different platforms. It would also help to include more features in the dataset, such as genre, publication year, page count, or how well-known the author is, because these factors might explain some of the variation in price that rating alone cannot. Another idea is to collect user reviews and run basic text analysis to see whether certain keywords appear more often in high-rated or low-rated books. Finally, the whole workflow could be turned into an automated pipeline that updates the dataset regularly and reruns the analysis. These additions would make the project deeper and allow for more meaningful conclusions about how online book markets behave.