5/5/2025

محمد حسام الدين محمد محمد

Facaluty of Computers & data science

**Preparation :**

|  |  |
| --- | --- |
| **محمد حسام الدين محمد محمد** | 23011471 |
| **محمد ياسر محمد سليمان النفراوي** | 23011507 |
| **محمد أيمن السيد أحمد حامد** | 23011465 |
| **محمد جهاد محمد أحمد علام** | 23011467 |
| **محمد أحمد سعد السيد يوسف** | 23011 |

Books Price Analysis Web Scraping.

[Final Report]

**Book Price Analysis Project — Full Report**

**Introduction**

The **Book Price Analysis** project is a comprehensive data-driven study focused on understanding the pricing dynamics of books sold across different online platforms. In today’s digital economy, **book buyers** and **sellers** face rapidly changing price landscapes, influenced by factors such as demand, availability, and store-specific marketing strategies.

The project's main objective is to develop an end-to-end pipeline that:

* Collects book data from online stores
* Cleans and standardizes this data
* Analyzes trends, patterns, and anomalies
* Visualizes findings to derive actionable insights
* Stores the processed data into a database for future querying and extension

By completing this project, we aim to empower users — both individual buyers and businesses — with better decision-making tools regarding book purchasing and pricing strategies.

The project execution is divided into **four major phases**:

* **Data Extraction**
* **Data Cleaning and Regular Expressions**
* **Data Analysis and Visualization**
* **Database Connection and Storage**

Each stage is critical to ensuring that the final insights are both accurate and valuable.

**1. Data Extraction**

Data extraction is the foundation of the project. Without **high-quality raw data**, all subsequent steps would suffer in accuracy and reliability.

**Approach:**

* **Web Scraping** was employed to gather real-time book data from online stores.
* **Websites scraped** : [**Books To Scrape**](https://books.toscrape.com)**,** [**Open Library**](https://openlibrary.org)**,** [**Google Books API**](https://www.googleapis.com)
* Python libraries used: requests & BeautifulSoup.

**Steps Taken:**

* **HTTP Requests**:
  + Used the requests library to send GET requests to the web pages containing book listings.
  + Ensured that the connection was successful by checking for a 200 OK response.
  + Managed headers and session details to mimic a real user and avoid being blocked.
* **HTML Parsing**:
  + Employed BeautifulSoup to parse the HTML content.
  + Carefully inspected the DOM structure to locate:
    - Book titles
    - Prices
    - Store names
    - Book categories (genres)
    - Availability statuses
* **Category Scraping**:
  + Extracted book categories either directly from the listing or from breadcrumb navigation structures.
  + This allowed later analysis of price differences between genres (e.g., Fiction vs. Academic Books).
* **Pagination Handling**:
  + Detected whether more book listings were available through pagination.
  + Implemented loops to dynamically scrape across multiple pages.
* **Data Structuring**:
  + Stored each scraped book’s information in a Python dictionary format immediately during scraping.
  + All book dictionaries were compiled into a list for easier conversion to a DataFrame.
* **Error Handling**:
  + Included try-except blocks to catch missing elements.
  + Default values such as "Unknown Category" or "N/A" were used to maintain a uniform dataset structure.

**Challenges:**

* Some stores embedded price information within scripts, requiring additional parsing logic
* Different sites presented similar information with varying levels of detail.
* Dynamic content loading (e.g., via JavaScript) made certain sites difficult to scrape traditionally. Despite these challenges, a **rich dataset** covering hundreds of books was successfully collected

**2. Data Cleaning & Regular Expressions**

Raw scraped data typically contains noise and inconsistencies. Before proceeding with any analysis, the data needed to be thoroughly cleaned.

**Techniques Applied:**

* **Price Cleaning**:
  + Prices were scraped as text strings that sometimes included currency symbols ($, €) or unnecessary text.
  + Applied **Regular Expressions** (re module) to extract only the numeric part and convert it into a floating-point number.
* **Text Standardization**:
  + Trimmed leading/trailing whitespaces.
  + Capitalized store names and categories uniformly to ensure consistency.
* **Handling Missing or Inconsistent Data**:
  + Some books lacked full information (missing authors or categories).
  + Replaced missing entries with placeholder text like "Unknown Author" to avoid analysis errors.
* **Duplicate Removal**:
  + Checked for and removed any duplicate book entries caused by repeated scraping.
* **Data Type Correction**:
  + Converted numeric fields like price into float types.
  + Standardized availability as either Available or Out of Stock.

**Importance of Cleaning:**

Without this cleaning stage, later visualizations and insights would be skewed or misleading. Cleaned datasets ensured high **data integrity** and **trustworthy analytics**.

**3. Data Analysis and Visualization**

In this section, we perform a detailed analysis of the books dataset using descriptive statistics and visualizations. Our goal is to better understand the distribution of book prices, ratings, and categories, and to uncover key patterns within the data.

**3.1 Basic Descriptive Statistics**

We began by calculating some key metrics:

* **Average Book Price**: [ $ 37]
* **Average Book Rating**: [2.94 out of 5]
* **Total Number of Books Analyzed**: [900 books]

These statistics provide an initial overview of the dataset and set the foundation for deeper analysis.

**3.2 Visual Analysis**

We generated several visual representations to better understand the data:

**3.2.1 Distribution of Book Prices**

A histogram (or boxplot) of book prices was created.

* Most books are priced within a specific range ($20–$50).
* A few books are significantly more expensive, acting as outliers.

**3.2.2 Distribution of Book Ratings**

A histogram of book ratings was plotted.

* The distribution is skewed toward higher ratings, with many books rated above 4.0.

**3.2.3 Top Rated Books**

A bar chart highlighted the highest-rated books.

* Titles with the best user feedback were identified, indicating popular and well-reviewed books.

**3.2.4 Books by Category**

A bar graph (or pie chart) showed the number of books per category.

* Certain categories (such as Fiction, Technology, or Science) were dominant, suggesting user preferences.

**3.2.5 Relationship Between Price and Rating**

A scatter plot was drawn to explore the relationship between price and rating.

* The plot suggested **little to no strong correlation** between how expensive a book is and its user rating.

**3.3 Key Findings**

* **Most books** are priced moderately, with few highly expensive outliers.
* **Book ratings** tend to be positive, with a majority rated 4.0 and above.
* **Categories** like **Programming** had the highest number of books.
* **Price and rating** are largely independent, meaning expensive books are not always better rated.

Through descriptive statistics and graphical analysis, we obtained valuable insights into the books dataset. Visualizations were particularly effective in revealing patterns that numerical summaries alone could not highlight.  
These findings can be used to inform pricing strategies, marketing campaigns, and further predictive analyses.

**4. Database Connection and Storage**

Finally, to ensure the data could be used in future applications, it was **saved systematically** into a database.

**Steps Taken :**

* **Database Setup**:
  + Created an **SQLite** database as a lightweight, portable solution.
* **Schema Design**:
  + Designed a table with fields:  
    Title, Price, Store, Category, Availability
* **Data Insertion**:
  + Inserted the entire cleaned dataset into the database using SQL commands executed through Python (sqlite3 module).
* **Advantages**:
  + Enables easy querying of historical data.
  + Makes the system scalable — more stores and new scraping sessions can append to the database.
  + Prepares the data for further projects like recommendation systems or dynamic dashboards.

**Summary**:  
The structured database enables **future expansion** and ensures that the work done today remains useful tomorrow.

**Conclusion**

The **Book Price Analysis** project demonstrates a full pipeline from **web data extraction** to **data cleaning**, **in-depth analysis**, **visualization**, and **database storage**.

Key achievements:

* Successfully extracted and cleaned a real-world dataset.
* Identified important pricing patterns and store differences.
* Created visually compelling reports and insights.
* Built a foundation for scalable, queryable data storage.

The project can be extended in future work by:

* Adding dynamic price tracking over time (monitoring price drops).
* Incorporating book reviews or ratings into the analysis.
* Expanding to additional stores or including international markets.
* Deploying a simple web dashboard to allow users to explore the data interactively.

This project highlights the **critical role of data-driven decision-making** in modern e-commerce environments.