Process of Crawling Amazon Book Data

1.Background

The primary goal is to detailed user review information on a wide range of books from Amazon, collecting multiple user reviews for each book to allow a robust, multi-dimensional analysis. The data will allow us to analyze user preferences and sentiment across various dimensions.

2.Target Data Fields:

Reviews of the books includes the following fields:

Id: Unique identifier for the book to facilitate data linking and referencing.

Title: Book title, providing a quick overview of the book's content.

Price: The current price of the book.

User\_id: Unique identifier for each reviewer, allowing for the analysis of different patterns.

profileName: Name of the reviewer for intuitive reference to the source of each review.

review/helpfulness: Helpfulness votes of the review, indicating the review's value to other users.

review/score: Star rating given by the user, used to determine overall rating trends.

review/time: Date of the review, enabling analysis of user feedback over time.

review/summary: Review title or summary, providing a quick snapshot of the user's opinion.

review/text: Detailed review content, offering insights into the user’s reading experience and feedback.

3.Data Storage and Structuring

Store the collected data in a CSV file (Books\_rating.csv) for easy access and subsequent processing using analysis tools like Excel, Python and so on.

4.Anti-Bot Measures

Since Amazon has strict anti-scraping measures, I need to simulate real user behavior to avoid triggering security mechanisms of Amazon. Using multiple User-Agent strings, random request intervals, and error handling strategies will reduce the risk of IP bans or other restrictions.

5.Result

Got about 1500 books and related 160000 reviews or so, containing information book title, username, review score, review text and so on. The data need to be cleaned for further analysis.

Process of Data Cleaning

After collecting the data we need, the next step is data cleaning. The main goal is to clean and format the book review data stored in Books\_rating.csv to prepare it for further analysis. The code addresses missing values, data type inconsistencies, and date formatting issues to ensure data accuracy and usability.

1.Process Overview

Data Loading: The code starts by reading the raw data from Books\_rating.csv using pandas. The encoding is set to UTF-8 to handle any special characters in the data.

Removing Unnecessary Columns: The Price column is dropped as it is not required for further analysis.

Data Type Conversion: Ensures that the review/helpfulness column is treated as a string. This is a crucial step to prevent data type errors in subsequent processing.

Handling Missing Values: Rows containing missing data in any column are removed to ensure data integrity.

Timestamp Conversion: The code defines a convert\_timestamp function to transform Unix timestamps in the review/time column into a readable YYYY/MM/DD format. In case of invalid timestamps (e.g., non-numeric values), the function returns None. After conversion, any row with None values in the review/time column is removed.

Saving Cleaned Data: The cleaned dataset is saved as Cleaned\_Books\_rating.csv for future analysis, with UTF-8 encoding maintained.

2.Error Handling

The convert\_timestamp function uses a try-except block to catch errors like invalid timestamps, returning None for problematic values.

This approach allows the code to handle inconsistencies without interrupting the process, enhancing its robustness.

3.Conclusion

This data-cleaning code effectively prepares Books\_rating.csv for analysis by ensuring consistency, handling missing values, and converting timestamps into readable dates. The result is a clean dataset that is ready for further analysis.