**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 reason why we collect user reviews is that they are a valuable source of information for understanding what readers think about a book. Reviews not only show how satisfied people are but also give helpful feedback about the content.

By collecting and analyzing this data, we can get a better idea of user preferences across various dimensions. We can look at several important factors:

**Sentiment Analysis**: We can analyze the tone of the reviews to see if they are positive, negative, or neutral. This helps us understand how readers feel about the book and what parts they like or dislike.

**Rating Trends**: We can track how ratings change over time. This lets us see which books become more popular or lose their appeal, helping us understand how public opinion shifts.

**Detailed Review Insights**: Reviews often mention what readers liked or didn’t like about a book. By studying these details, we can identify the book’s strengths and weaknesses, which can help authors and publishers improve their work.

**Reputation and Credibility**: Reviews play an important role in helping potential buyers decide whether to purchase a book. By looking at the overall feedback, we can understand how well a book is received and how much trust readers place in it.

**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**

Once the data is collected, it will be stored in a CSV file (Books\_rating.csv). This format is widely used due to its simplicity and compatibility with most analysis tools. The structured format will include columns for book titles, user IDs, ratings, review text, and other relevant information. This structured data can then be imported into tools like Excel, Python, or other data analysis platforms for further processing, cleaning, and analysis.

**4.Anti-Bot Measures**

Since Amazon has strict anti-scraping measures, we 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. This large dataset will provide valuable insights into user opinions and preferences. However, the collected data may include inconsistencies, missing values, or unnecessary information. Therefore, a data cleaning process will be needed to ensure the dataset is ready for further analysis.

**Process of Data Cleaning**

**1.Process Overview**

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. These are the key actions:

**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. If an invalid timestamp is encountered (like a non-numeric or incorrect value), the function returns None for that value. This prevents the process from stopping due to a single problematic row, and ensures that the rest of the data can still be processed without interrupting the process.

**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 cleaned data now is structured and formatted for deeper analysis, making it suitable for exploring trends and user insights. This step sets a strong foundation for the next stage.