Market Basket Analysis on Amazon Book Reviews Using FP-Growth Algorithm

Samet Cagan
Matriculation: 985869
Master in Data Science for Economics

Academic Year 2024/2025

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work, and including any code produced using generative AI systems. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

1 Introduction

The goal of this project is to apply market basket analysis techniques to the Amazon Book Reviews dataset to uncover patterns in user behavior and review content. The FP-Growth algorithm is used to efficiently discover frequently used item clusters without the need for candidate generation, making it suitable for large-scale datasets. Data preprocessing, including title normalization and stop word removal, prepared the data for meaningful pattern discovery. The ultimate aim is to generate insights that can support the development of a book recommendation system and provide a better understanding of reader preferences and language patterns.

2 Dataset Description

2.1 Dataset Selection

I use the Amazon Books Reviews dataset, which is publicly available corpus of user-generated reviews for books sold on Amazon. The dataset is widely used in research on recommender systems, text mining, and consumer behavior analysis.

2.2 Dataset Structure

The dataset has two main components:

1. **Book Details**: Provides descriptive metadata such as title, authors, publisher, publication date, categories, etc.
2. **Reviews**: it has user identifiers, book identifiers, rating scores, review text, review/helpfulness, etc.

Each review corresponds to a single user–book interaction. For this project, only attributes required for market-basket analysis are retained. Specifically, ”user-id” is used to define transactions, and ”titles” are used to define items. All other attributes, including review text, ratings, and detailed metadata, are excluded as they are not directly necessary for my purpose.

2.3 Interpretation for Market-Basket Analysis

In this project, the dataset is interpreted as follows:

- Each user corresponds to a single transaction (basket)
- Each transaction consists of the set of books reviewed by that user (item)
- Multiple reviews of the same book by the same user are treated as a single item occurrence

This interpretation enables the direct application of frequent itemset mining algorithms to discover associations among books frequently reviewed together by the same users.

3 Data Organization and Representation

3.1 Transformation into Transactional Data

The dataset represents each review as a separate row. To support market basket analysis, this structure groups the review records by user identifier. Then, the related book titles are combined into a single cluster per user, transforming it into a transactional representation. As a result, each user is represented by a single transaction containing all the different books they have reviewed.

3.2 Transaction Definition

Formally, let $\mathcal{U}$ be the set of users and $\mathcal{B}$ be the universe of all book titles. Each transaction T_u corresponding to user $u \in \mathcal{U}$ is defined as:

$$T_u = \{b_1, b_2, \dots, b_k\} \subseteq \mathcal{B}$$

where b_i represents a distinct book reviewed by user u. The resulting dataset $\mathcal{D}$ consists of a collection of such transactions and forms the basis for subsequent frequent itemset and association rule analysis.

4 Model

4.1 Market-Basket Model

In this project, I use a market-basket model to represent the data as a set of baskets, where each basket contains a group of related items. The goal is to find items that often appear together across many baskets. In this case, the items are books and the baskets are users. Each basket includes the books reviewed by a particular user.

The Amazon Books review dataset fits this approach well and is commonly used in recommendation systems. By finding frequent itemsets, I aim to discover relationships between books that reflect common reading interests among users.

To make the analysis manageable and reduce noise, I include only users who have reviewed at least a minimum number of books and exclude users with very large numbers of reviews. This decision is based on the strong imbalance in user activity, where most users write only a few reviews while a small number of users write many. Filtering helps prevent these extreme cases from affecting the results.

I use the FP-Growth algorithm to find frequent itemsets, as it is efficient and scalable for large datasets. Minimum support and confidence thresholds are applied to limit the number of patterns and keep only meaningful associations. These frequent itemsets and rules are then used to produce book-to-book recommendations.

4.2 Frequent Itemsets

An itemset $I \subseteq \mathcal{B}$ is defined as frequent if it appears in at least a predefined proportion of baskets, known as the support threshold. Let $N = |\mathcal{D}|$ be the total number of transactions. The **Support** of an itemset I is defined as:

$$\text{Support}(I) = \frac{|\{T \in \mathcal{D} \mid I \subseteq T\}|}{N}$$

The itemset I is considered *frequent* if $\text{Support}(I) \geq s_{min}$, where s_{min} is the minimum support threshold defined for the study.

4.3 Scope of Analysis

The project addresses both frequent itemset mining and association rule analysis. From frequent itemsets, association rules of the form $X \to Y$ are generated, where X and Y are disjoint sets of books ($X, Y \subset \mathcal{B}$ and $X \cap Y = \emptyset$).

The strength of an association rule is evaluated using **Confidence** and **Lift**.

- **Confidence** measures the conditional probability that a user reviews Y given that they have reviewed X:

$$\text{Confidence}(X \to Y) = P(Y|X) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

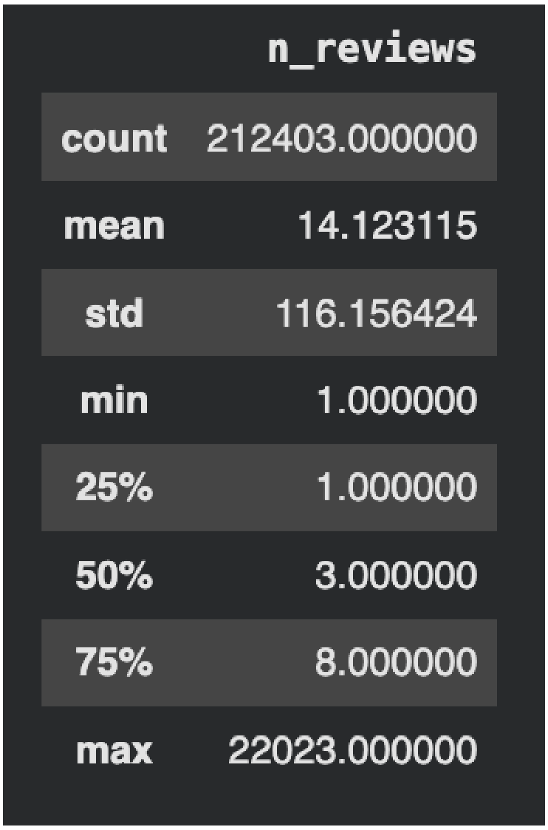
- **Lift** assesses whether the observed association exceeds what would be expected if X and Y were independent:

$$\text{Lift}(X \to Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X) \times \text{Support}(Y)}$$

A lift value greater than 1 indicates a positive association between the antecedent and the consequent.

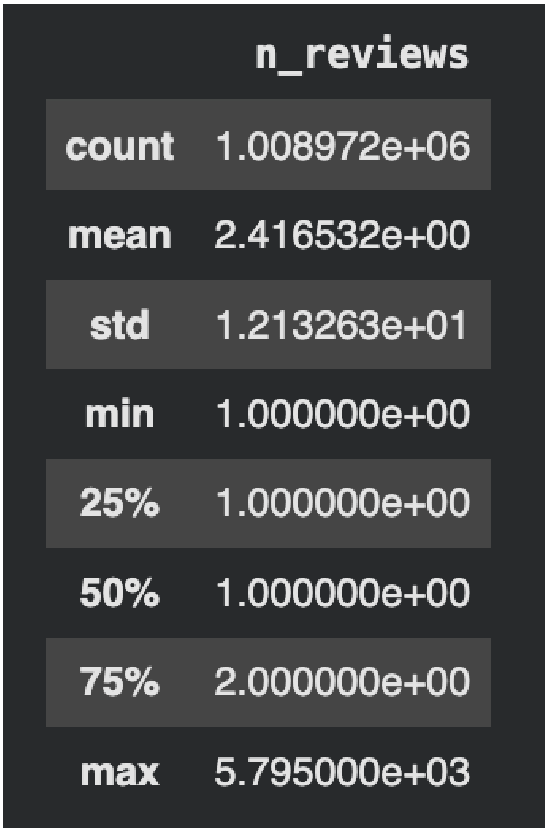
5 Data Exploration and Pre-processing

I first explored the dataset to understand its size and structure. The dataset contains 212,403 unique book titles and 1,008,972 unique users. The distribution of reviews is highly skewed. On average, a user writes 2.42 reviews, but the median number of reviews per user is only one. This shows that most users contribute very few reviews, while a small number of users are much more active.



	n_reviews
count	212403.000000
mean	14.123115
std	116.156424
min	1.000000
25%	1.000000
50%	3.000000
75%	8.000000
max	22023.000000

Figure 1: Review statistics per book



	n_reviews
count	1.008972e+06
mean	2.416532e+00
std	1.213263e+01
min	1.000000e+00
25%	1.000000e+00
50%	1.000000e+00
75%	2.000000e+00
max	5.795000e+03

Figure 2: Review statistics per user

The average number of reviews per book is 14.12, while the median is three, indicating that many books receive only a small number of reviews. Because of the small median basket size and the large number of rare items, the resulting transaction data is very sparse.
To keep the analysis consistent, I treat each basket as a set rather than a multiset, meaning that repeated reviews of the same book by a user are counted only once.

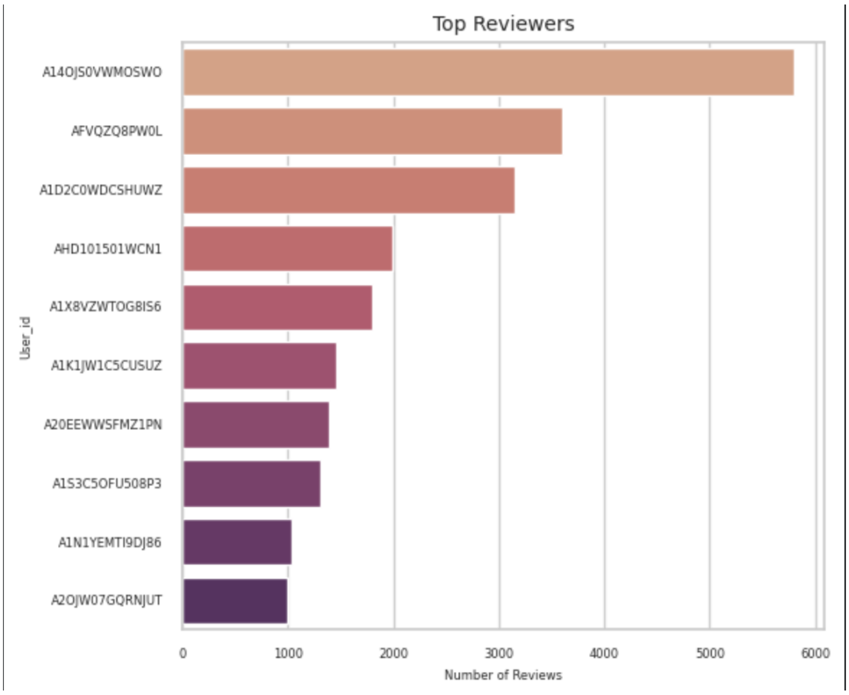


Figure 3: top reviewers-Userid

5.1 Implications for Frequent Itemset Mining

Because the data is very sparse, only a small number of itemsets can meet reasonable support thresholds. At the same time, the long-tail distribution of book popularity means that very low support values are dominated by a few extremely popular books. As a result, choosing an appropriate support threshold is important to balance how many patterns are found and how meaningful those patterns are.

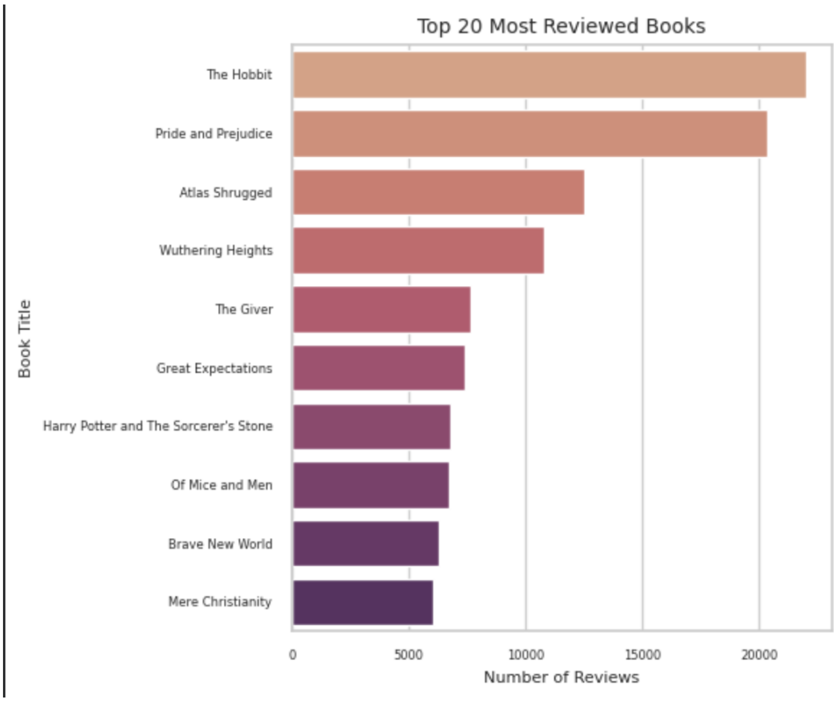


Figure 4: most reviewed books

5.2 Title Normalization and Canonicalization

To avoid splitting support counts across slightly different versions of the same title, I applied a basic title normalization step. This included converting titles to lowercase and removing punctuation and stop words. The cleaned version of each title is referred to as TitleClean and is used as a canonical representation. After normalization, the number of unique titles was reduced from 212,403 to 205,589, indicating that many near-duplicate titles were successfully merged.

6 Algorithms and Implementation

6.1 Transaction Construction

Transactions are constructed by grouping review records by user identifier and collecting the set of distinct cleaned book titles reviewed by each user. This representation treats each user as a single basket and ensures that repeated reviews of the same book by a user do not inflate item frequencies. Users who reviewed fewer than two distinct books are removed ($|T_u| < 2$), since singleton baskets cannot contribute to frequent itemsets or association rules involving multiple items. After applying these preprocessing and filtering steps, the final dataset used for market-basket analysis consists of 255,888 user baskets.

6.2 FPGrowth

Given the large size and sparsity of the dataset, I use the FP-Growth (Frequent Pattern Growth) algorithm, implemented through Apache Spark MLlib. FP-Growth is well-suited for large-scale data because it avoids generating candidate itemsets explicitly, which significantly reduces both memory usage and computation time. Instead, it compresses the transaction data into a compact tree structure and recursively extracts frequent patterns from this representation. Compared to the Apriori algorithm, FP-Growth requires fewer passes over the data and scales more efficiently, making it a practical choice for this dataset.

6.3 FP-Growth Configuration

The algorithm is configured with the following thresholds:

- **Minimum Support (s_{min}):** 0.0005
- **Minimum Confidence (c_{min}):** 0.05

These thresholds ensure that retained itemsets are shared by a substantial number of users while filtering out weak associations.

7 Experimental Evaluation

7.1 Frequent Itemset Results and Similarity Filtering

Analysis focuses primarily on itemsets of size two ($|I| = 2$). Raw results are often dominated by different editions of the same book (e.g., "The Hobbit" and "The Hobbit: Illustrated"). To mitigate this, a lexical similarity filter based on **Jaccard Similarity** is applied.

Let S_A and S_B be the sets of words (tokens) comprising the titles of book A and book B. The Jaccard similarity $J(A, B)$ is defined as:

$$J(A, B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|}$$

Itemsets containing books where $J(A, B) > 0.5$ are removed to filter out trivial self-associations.

7.2 Association Rule Analysis

The highest-ranking rules exhibit consistently high lift values, often exceeding several hundred. Prominent patterns include:

- Strong associations among works by the same author.
- Co-reading within established literary series (e.g., *The Two Towers* → *The Return of the King*).
- Thematic associations among related genres.

7.3 Series Continuation Effects

Rules reflecting series progression display moderate-to-high confidence and extremely high lift. For example, in a trilogy sequence $A \rightarrow B$, the lift is calculated as:

$$\text{Lift}(A \rightarrow B) \gg 1$$

This confirms strong dependency between sequential volumes.

- The Two Towers → The Return of the King
- The Golden Compass → The Subtle Knife
- The Subtle Knife → The Amber Spyglass

These rules exhibit both moderate-to-high confidence and extremely high lift, reflecting a strong sequential reading tendency among users. This phenomenon is well-documented in market-basket analysis and serves as an important validation signal for the correctness of the basket construction and mining process.

7.4 Residual Duplication

Despite applying several preprocessing and normalization steps, some residual duplication remains in the data. This issue mainly arises from different editions or formats of the same book, which may still appear as separate items after cleaning. Although these near-duplicate titles are not fully removed through lexical filtering alone, their presence does not significantly affect the overall results. Instead, it highlights a common limitation when working with real-world book data, where precise edition-level matching is difficult without external identifiers such as ISBNs.

7.5 Limitations

While the proposed approach is effective, there are several limitations that should be acknowledged. First, association rules remain sensitive to residual duplication at the edition level, which can lead to associations between closely related versions of the same book. Second, semantic similarity between books is captured only through shared titles and co-review patterns, rather than deeper semantic content. Finally, user behavior is modeled in a binary way, where a book is either reviewed or not reviewed, without considering rating intensity or the order in which books were reviewed. These limitations suggest several possible extensions, such as incorporating semantic embeddings, temporal information, or weighted transactions based on user ratings.

8 Summary

In this project, I explored the application of market-basket analysis to a large-scale Amazon Books review dataset. By modeling users as transactions and books as items, I applied frequent itemset mining and association rule analysis to uncover common co-reading patterns. The FP-Growth algorithm proved to be well suited for this task due to its scalability and efficiency on sparse data. Careful preprocessing, including title normalization and basket-size filtering, was essential to make the analysis computationally feasible. Although the results are influenced by data sparsity and residual duplication, the discovered patterns provide meaningful insights into shared reading behavior and demonstrate the practical value of frequent pattern mining for recommendation-oriented analysis.