Algorithms for Massive Data: Market-Basket Analysis on the Amazon Books Reviews Dataset Using PySpark FP-Growth

Alina Imansakipova

October 2025

Declaration of Authorship

I/We declare that this material, which I/We now submit for assessment, is entirely my/our 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/our work, and including any code produced using generative AI systems. I/We 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/us or any other person for assessment on this or any other course of study.

Abstract

This project implements a scalable system for discovering frequent itemsets and association rules from large datasets. Using the Amazon Books Reviews dataset from Kaggle, two distinct market-basket analyses are performed: (1) identifying co-reviewed books by the same users, and (2) discovering co-occurring words within user reviews. The system is implemented entirely in PySpark and employs the FP-Growth algorithm for distributed frequent pattern mining. The results demonstrate both the interpretability and scalability of Spark-based association mining for massive text and transactional data.

1 Introduction

Market-basket analysis is a classical data mining technique used to uncover relationships between items in large transactional datasets. It aims to identify sets of items that frequently occur together, as well as association rules that describe these co-occurrences in probabilistic terms. Traditional algorithms such as Apriori become computationally expensive on large-scale data due to repeated

candidate generation and multiple passes over the dataset. The FP-Growth (Frequent Pattern Growth) algorithm provides an efficient alternative, avoiding explicit candidate enumeration and relying on a compressed data representation (the FP-tree). In this project, a scalable implementation of FP-Growth is developed using Apache Spark, which enables distributed processing of millions of records. The system is applied to the Amazon Books Reviews dataset, where each review contains user identifiers, product identifiers, and textual feedback. Two perspectives are explored:

- User–Book Analysis — Each user is treated as a basket of reviewed books.
- Word–Review Analysis — Each review is treated as a basket of words.

The objective is to extract patterns such as:

- Which books tend to be reviewed together by the same users.
- Which words tend to co-occur in the text of reviews.

2 Dataset Description

2.1 Source

The dataset used is the Amazon Books Reviews dataset, publicly available on Kaggle under the CC0 license. It contains approximately three million book reviews, including user information, product identifiers, ratings, timestamps, and full review texts. The dataset is automatically downloaded during execution using the Kaggle API.

2.2 Selected Columns

The dataset's schema includes multiple fields such as: Id, Title, Price, User id, review/score, review/time, review/summary, and review/text. Only the following columns are used in this analysis:

- Id as book id – unique book identifier
- User id as user id – unique reviewer identifier
- review/text as review text – full review text

By selecting only these necessary columns (Id, User id, review/text) in an early step, column pruning was performed, which significantly reduced the I/O and memory footprint for subsequent processing steps, adhering to massive data best practices.
An example of a cleaned dataset is below:

```
+----------+---------------+-------------------------------------------------------------------------------------------------------------+
|   book_id|        user_id|                                                                                                  review_text|
+----------+---------------+-------------------------------------------------------------------------------------------------------------+
|1882931173| AVCGYZL8FQQTD|This is only for Julie Strain fans. It's a collection of her photos -- about 80 pages worth with ...|
|0826414346|A30TK6U7DNS82R|I don't care much for Dr. Seuss but after reading Philip Nel's book I changed my mind--that's a g...|
|0826414346|A3UH4UZ4RSVO82|If people become the books they read and if "the child is father to the man," then Dr. Seuss (The...|
|0826414346|A2MVUWT453QH61|Theodore Seuss Geisel (1904-1991), aka &quot;Dr. Seuss,&quot; was one of the most influential wri...|
|0826414346|A22X4XUPKF66MR|Philip Nel - Dr. Seuss: American IconThis is basically an academic overview of Seuss poetry, art,...|
+----------+---------------+-------------------------------------------------------------------------------------------------------------+
```

3 Data Preparation and Organization

Two analytical views of the dataset were constructed:

1. User–Book baskets
2. Word–Review baskets

3.1 User–Book Baskets

Each user was treated as a transaction containing all books they have reviewed. The following steps were applied:

1. Filter out missing user or book identifiers.
2. Group by user id and aggregate all reviewed book id values into a set.
3. Retain only baskets with at least two items.

Result: 308,676 baskets (users who reviewed more than or equal to 2 books). An example is provided below

```
Number of user baskets: 308676
+--------------+-------------------------------------------------------------------------------------------------------------------+
|       user_id|                                                                                                              items|
+--------------+-------------------------------------------------------------------------------------------------------------------+
|A1005YJDO9VCIY|                                                                                                [0977390403, 1590830591]|
|A1006V961PBMKA|                                                                                                [B000N6OSCM, 0742516814]|
|A1018G2FPJBJ0S|                                                                                                [B000KW0HGK, B0006C2DE8]|
|A1036ICK1O363W|                                                                          [1559390921, 0553354566, 15593918BX, 1932100385]|
|A10AKE9TAADHVV|[1569553688, 1892112604, B000PXIPT0, B000IOIN6G, B0007HE34U, 1559716843, 0679886524, 1582381496, B0007160ZW, B000FQ4J...|
+--------------+-------------------------------------------------------------------------------------------------------------------+
```

Each basket is a set of book identifiers.

3.2 Word–Review Baskets

Each review was treated as a transaction containing all unique meaningful words from the review text. The process includes:

1. Lowercasing the text.
2. Tokenization using a regular expression to split on non-word characters.
3. Removal of stopwords using Spark's built-in StopWordsRemover.
4. Deduplication of tokens per review.
5. Filtering out baskets with fewer than two words.

Due to resource constraints, a 3% subsample of reviews was used for text-based analysis.
Result: 90,284 baskets (reviews after preprocessing). Each basket contains the distinct words from one review. The example of the dataset is below.

```
Number of word baskets: 90284
+----------------------------------------------------------------------------------------------------------------------------------------------------+
|                                                                                                                                               items|
+----------------------------------------------------------------------------------------------------------------------------------------------------+
|[self, published, book, want, know, read, paragraphs, 5, star, reviews, must, written, ms, haddon, family, friends, p...|
|[chance, read, book, sure, bothered, purchasing, store, sourdough, lore, interesting, order, make, seem, authenticall...|
|[book, worth, keep, collection, advise, sourdough, ruth, also, told, picture, past, 100, years, ago, alaska, stand, m...|
|            [gave, detailed, vision, day, life, commune, still, existence, today, interesting, quick, delightful, read]|
|[dr, baker, one, great, 20th, century, metaphysicians, like, emmet, fox, ernest, holmes, thomas, troward, understood,...|
+----------------------------------------------------------------------------------------------------------------------------------------------------+
```

4 Methodology

4.1 Algorithm

Frequent itemset mining and association rule extraction were performed using the **FP-Growth algorithm**, implemented in PySpark's MLlib (pyspark.ml.fpm). FP-Growth compresses transactions into an FP-tree and recursively mines frequent patterns without explicit candidate generation, making it efficient for large datasets. Two sets of parameters were used:

- MIN SUPPORT – minimum fraction of transactions containing an itemset: 0.01 for Books and 0.04 for Words. The 0.04 threshold for words was determined empirically to manage the large vocabulary size; initial attempts with lower values (0.01 and 0.025) resulted in OutOfMemoryError exceptions, even with **32**GB of dedicated Spark memory.
- MIN CONFIDENCE – minimum conditional probability for association rules – 0.5 for Books and Words

These thresholds balance interpretability and computational feasibility. Lower support is used for the sparse user–book matrix, while higher support is necessary for frequent word co-occurrence.

5 Results

5.1 Frequent Itemsets: User–Book Baskets

The model identified recurring combinations of books reviewed by the same users. Several book identifiers appear consistently together. These likely represent editions or volumes of the same titles, reflecting consistent co-review behavior among users. The frequent itemsets also include single items (individual books or words), which represent globally popular elements in the dataset and serve as the foundation for forming larger co-occurring patterns. Figure 1 reports the top 20 frequent itemsets of books co-reviewed by the same users. These represent recurring combinations of items appearing in user baskets.
Confidence = 1.0 indicates deterministic co-occurrence of the consequent when the antecedent appears. High lift values (more than 80) show strong correlations between specific books, suggesting a shared readership or near-identical product entries.
The consistently high Lift values, often exceeding 80, indicate that the co-occurrence of the books is 80 times more frequent than what would be expected

if the books were reviewed independently. This confirms a highly dependent,non randomrelationship useful for deterministic recommendation models (Figure 2).

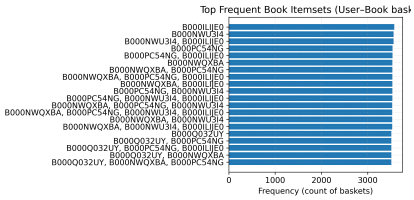

Figure 1: Top 20 frequent itemsets of books co-reviewed by users, showing recurring combinations with high support.

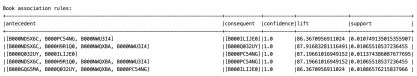

Figure 2: Association Rules: User–Book Baskets

5.2 Frequent Itemsets: Word–Review Baskets

The most common co-occurring words in reviews are represented in Figure 3. These frequent patterns reflect natural review structure and language use. Phrases like "good book" and "like book" are highly represented.

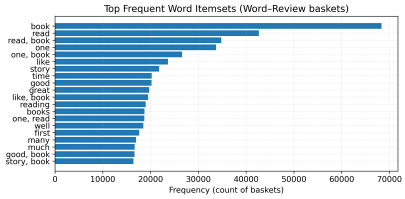

Figure 3: Top 20 frequent co-occurring word itemsets extracted from review texts

Word association rules:
[Stage 45:> (0 + 1) / 1][Stage 46:> (0 + 1) / 1]

antecedent	consequent	confidence	lift	support
[good, like, read]	[book]	0.9097744360902256	1.2019209410142075	0.04556732089849807
[first, like, read]	[book]	0.9065442020665901	1.1976534151711324	0.04372867839262771
[think, good]	[book]	0.9047261815453863	1.1952515924237862	0.04007354570023482
[think, one, read]	[book]	0.9041095890410958	1.194436999912002	0.04093748615480041
[much, like, read]	[book]	0.9040537299112497	1.1943632033144655	0.04174604581099641

Figure 4: Word association rules

The consequent "book" dominates association rules, indicating that words related to reading and evaluation almost always co-occur with "book." This confirms the thematic coherence of review texts (Figure 4).

6 Scalability and Replicability

The entire pipeline is implemented in PySpark, ensuring scalability across larger datasets and distributed environments. All transformations (groupBy, collect set, tokenization, FP-Growth) are executed as Spark operations, allowing horizontal scaling by simply adjusting computational resources. Caching and column pruning minimize repeated computation.

Managing High Dimensionality

The primary scalability challenge was encountered during the Word-Review Analysis due to the massive vocabulary size. Initial attempts using lower support values (0.01) led to java.lang.OutOfMemoryError crashes, even after allocating a substantial **32**GB of memory to the Spark driver (spark.driver.memory).

The solution demonstrates that effective scaling relies on algorithmic tuning alongside resource provision. The necessary tuning was:

- Memory Provisioning: Allocation of **32**GB to the Spark driver.
- Parameter Tuning: Incremental adjustment of MIN SUPPORT WORDS from the failing 0.01 and 0.025 values to the stable 0.04 threshold.

This process successfully pruned the exponentially growing FP-Tree, making the intensive computation feasible. The global variable USE SUBSAMPLE FOR WORDS enables efficient testing on subsets while preserving full-scale reproducibility. All outputs (frequent itemsets and rules) are exported as CSV files for verification.

7 Discussion

The results confirm that FP-Growth is an effective approach for uncovering both behavioral and linguistic associations in large-scale datasets.

- **For user–book baskets**, the algorithm identified deterministic rules among certain book identifiers, likely representing multiple editions or series.
- **For word–review baskets**, the model captured semantic regularities of language use, with strong associations between evaluative adjectives (e.g., good, great) and the term book.

The approach is robust, interpretable, and reproducible. It also demonstrates the feasibility of large-scale text mining using distributed FP-Growth.
The stable 4% minimum support yielded high-confidence rules that, while confirming semantic coherence, were still dominated by generic words related to reading (book,read,one). To discover more novel, genre-specific word associations, a critical next step would be to introduce a custom, domain-specific stop word list (including terms like book and read) and re-run the algorithm with a lower support threshold. This would shift the focus from common transactional structure to latent topical patterns.

8 Conclusions

This project presented a scalable implementation of market-basket analysis on the Amazon Books Reviews dataset using PySpark.
Two perspectives—users and words—were analyzed using FP-Growth to uncover frequent patterns and association rules.
Key outcomes:

- Construction of over 300,000 user-level baskets and 90,000 word-level baskets.
- Identification of meaningful itemsets and strong association rules in both transactional and textual contexts.
- Demonstration of efficient and scalable computation using distributed algorithms.

Because the entire workflow is implemented in PySpark using distributed transformations, the same codebase can seamlessly scale to much larger datasets without modification.