Link Analysis on Amazon Book Reviews using Graph-Based PageRank

Melany Yohana Gomez Herrada
Student ID: 39351A
Program: Data Science for Economics (DSE)
Course: Algorithms for Massive Data

December 2025

Declaration

1 Introduction

The objective of this project is to identify the most influential and central books within a large collection of user-book interactions from the Amazon Books dataset. The task is modeled as a graph-based link analysis problem, where nodes represent books and edges indicate co-occurrence — i.e., books rated by the same users.

We apply the PageRank algorithm to identify highly connected books within this network. This approach is particularly suited for large datasets, as it allows scalable and parallel computation using Apache Spark and GraphFrames.

This report presents the methodological approach, preprocessing steps, implementation details, and a discussion of the experimental results.

2 Dataset Description

The dataset used in this study is the **Amazon Books Reviews** dataset, available on Kaggle

Two CSV files were considered:

- `Books_rating.csv`: containing user IDs and titles of books rated.
- `books_data.csv`: containing metadata such as title, authors, and publishers.

From the rating file, we extracted only the columns:

- `User_id`
- `Title`

This subset forms the basis for building the interaction graph between books.

3 Data Organization and Preprocessing

To prepare the data for graph construction, the following steps were taken:

- **Loading into Spark DataFrames** with schema inference.
- **Deduplication**: all duplicate (user, book) pairs were removed.
- **Null handling**: rows with missing values were dropped.
- **Sampling**: a 5% sample of the data was used during development to reduce computation time, ensuring reproducibility via a fixed seed.
- **Grouping by user**: each user's set of rated books was collected into a list.

This transformation enabled us to model a **co-purchase graph**, where books are connected if they have been rated by the same user.

4 Graph Construction

From the cleaned and grouped data, we constructed the co-occurrence graph as follows:

- **Nodes**: each unique book title became a vertex.
- **Edges**: for each user, all pairs of books they rated were generated (combinations of 2).
- **Edge weights**: number of users who rated both books in the pair.
- **Edge filtering**: only edges with weight ≥ 2 were retained to remove weak/noisy connections.

This results in a **weighted undirected graph** of co-rated books. The graph was implemented using `GraphFrames`, an extension of Spark's GraphX API that supports DataFrame-based graph processing.

5 PageRank Algorithm and Implementation

To rank the most central books, we applied the PageRank algorithm, which assigns scores to nodes based on the structure of incoming links.

Implementation Details

- **Tool**: `GraphFrames`' `pageRank()` method.
- **Parameters**:
  - `maxIter = 10`: number of iterations.
  - `resetProbability = 0.15`: teleportation factor.

The resulting PageRank values reflect how frequently a book appears in co-rating relationships with other well-connected books, an indicator of influence or centrality in the reading ecosystem.

6 Scalability Considerations

The solution was applied with scalability in mind. The entire pipeline is implemented in **Apache Spark**, which supports distributed computation across large datasets. Instead of performing local or sequential operations, all transformations are executed using `Spark DataFrames`, ensuring efficient parallel processing and avoiding memory bottlenecks. Functions such as `UDFs`, `explode()`, and `collect_set()` are used to parallelize the generation of book pairs across the cluster.

To reduce memory consumption and improve execution time, only the essential columns are preserved throughout the pipeline. While a 5% sample of the dataset was used during development to facilitate rapid experimentation, the same architecture and logic can be seamlessly applied to the full dataset or even larger data sources. Overall, the system is ready for deployment in a distributed environment and is capable of scaling to millions of interactions.

7 Results of PageRank Centrality Analysis

We ran the PageRank analysis on a 5% sample of the dataset (approx. 106,000 rows) for testing purposes. The pipeline extracted a graph of co-rated books and identified central titles based on the computed PageRank scores.

Top 10 Books by PageRank

Title	PageRank Score
Wuthering Heights	4.94
Little Women	4.42
To Kill a Mockingbird	4.29
Moby-Dick	3.92
The Catcher in the Rye	3.64
The Lion, the Witch and the Wardrobe	3.38
The Lord of the Rings Trilogy	3.30
The Count of Monte Cristo	3.19
Jane Eyre	2.86
Slaughterhouse-Five	2.84

The results obtained through the PageRank analysis reveal that classical literary works such as *Wuthering Heights*, *Little Women*, and *To Kill a Mockingbird* consistently rank among the most central books in the co-rating graph. These books appear to function as **hubs** within the user interaction network, as they are frequently co-rated alongside a diverse range of other titles.

This pattern suggests that well-known literary works act as common touchpoints among users with otherwise diverse reading preferences, making them highly influential in the structure of the reading ecosystem. As such, they are ideal candidates for recommendation strategies or promotion in collaborative filtering systems.

The PageRank algorithm proved effective even without access to explicit rating values. By relying solely on co-occurrence information, the system was able to surface books with broad connectivity and impact, reinforcing the value of structural analysis in user behavior data.

8 Conclusion

This project presented a scalable, graph-based approach to link analysis using the Amazon Books Reviews dataset. By constructing a co-rating graph and applying the PageRank algorithm with Apache Spark and GraphFrames, we successfully identified books that hold a central role in the user-book interaction network.

The results show that centrality in the co-rating graph aligns closely with cultural prominence and broad readership, as evidenced by the consistent presence of canonical literary works in the top rankings. This confirms that structural relationships alone, without textual content or rating scores, can provide meaningful insights into user behavior and content influence.