Similarity Review Detection Using Jaccard and Cosine Similarity on Amazon Book Reviews

Aidana Akkaziyeva

June 9, 2025

1 Introduction

In this project, we address the problem of detecting semantically similar or duplicate book reviews within the Amazon Books Review dataset. Due to the scale of the data, we explore approximate similarity search techniques, comparing two locality-sensitive hashing (LSH) approaches: MinHashLSH based on Jaccard similarity and BucketedRandomProjectionLSH based on Cosine-like similarity.

2 Data Preprocessing Pipeline

2.1 Data Acquisition and Initial Preparation

To ensure reproducibility and access to a large-scale dataset, we utilized the Kaggle API to download the Amazon Books Reviews dataset. After downloading, the dataset was extracted and prepared for exploration.

2.2 Filtering and Cleaning for Relevance

Given the broad scope of the dataset (around 3 million records), it was essential to narrow the focus to a specific domain. Our analysis targeted cooking-related books, so we first retained only columns relevant to this goal: book title, author(s), publication date, and category tags.

We then identified all books categorized under "Cooking" (case-insensitive) and extracted them into a separate subset. To ensure data quality, we addressed potential redundancies by removing duplicate entries. Initially, duplicates were removed based on the combination of authors and publication dates, followed by a secondary deduplication based solely on the title.

After isolating the relevant books, we focused on user-generated ratings and reviews. We loaded the corresponding ratings data, keeping only the information necessary for textual analysis and user behavior—namely, review text, book title, user ID, and review time. Reviews with identical user ID and timestamp were considered duplicates and removed. We also eliminated null entries to preserve data integrity.

2.3 Integration and Final Refinement

Once both datasets—cooking books and their corresponding reviews—were cleaned, we merged them on the book title to create a unified dataset. This integration step ensured that each review was associated with a verified cooking book.

Further refinement involved validating the user identifiers. We retained only those user IDs that were entirely uppercase and free of spaces, assuming these conformed to the platform's standard formatting. This step likely helped filter out malformed or corrupted entries.

The final dataset—containing only unique, relevant, and well-formed entries for cooking books and their user reviews—was exported as a CSV file for subsequent analysis.

Summary Table

Step	Description
1	Import necessary libraries and prepare the programming environment
2	Download and extract the Amazon Books Reviews dataset using the Kaggle API
3	Load the books metadata and explore key attributes
4	Filter books by category to retain only cooking-related titles
5	Clean the dataset by removing duplicate and irrelevant records
6	Load and clean ratings data by removing duplicates and nulls
7	Merge books and reviews datasets to ensure consistent, relevant data
8	Validate user identifiers and remove incorrect User Ids

Table 1: Summary of the data preprocessing pipeline

3 Dataset Description

As a result of the previous preprocessing stages, each review contains a `Title`, `User_id`, and `review/text`. A representative subset of the dataset was used during development for faster iteration.

4 Data Organization and Preprocessing

The preprocessing pipeline included the following steps:

1. **Text normalization:** All text was lowercased and punctuation removed.

2. **Tokenization:** Text was split into individual words.

3. **Stopword removal:** A combination of standard stopwords and domain-specific ones (e.g., `"book"`, `"read"`, `"recipe"`) were removed.

4. **Filtering:** Reviews with fewer than 6 meaningful tokens were discarded.

5. **Shingling:** For Jaccard-based similarity, word-level bigrams were generated.

Each processed review was associated with a unique `Review_ID` for downstream pairwise similarity computation.

5 Algorithms and Implementation

5.1 Approach 1: MinHashLSH (Jaccard Similarity)

This approach is based on set similarity. Each review is represented as a set of bigrams (two-word sequences). These sets are transformed into fixed-size vectors using the HashingTF technique.

We then applied the MinHashLSH algorithm to detect reviews with high overlap in their bigram sets, measured by the Jaccard similarity. Pairs of reviews with a Jaccard distance below 0.8 were considered similar.

This method is particularly good at identifying near-duplicates or reviews with high word-pair overlap, even if the word order or syntax differs slightly.

5.2 Approach 2: TF-IDF + BucketedRandomProjectionLSH (Cosine-like Similarity)

In the second approach, each review was vectorized using TF-IDF. This method weighs each word based on how frequently it appears across the corpus, giving more importance to unique words that better characterize a review.

We then used BucketedRandomProjectionLSH to perform approximate similarity joins based on Euclidean distance, which, on TF-IDF vectors, approximates cosine similarity. This method is more sensitive to the meaning of words and the uniqueness of terms in a review. It can distinguish between reviews that use similar vocabulary but describe different aspects of a book.

5.3 Key Differences

Step	**Approach 1 (MinHashLSH)**	**Approach 2 (TF-IDF + BRP-LSH)**
Shingling	Word-level bigrams (n=2)	Single words (no n-grams)
Vectorization	HashingTF on bigrams	HashingTF + IDF (TF-IDF)
LSH Method	MinHashLSH (Jaccard similarity)	BucketedRandomProjectionLSH (Euclidean/Cosine)
Similarity	Jaccard distance on sets of shingles	Cosine-like similarity on TF-IDF
Threshold	Jaccard distance < 0.8	Euclidean distance < 5.0

Table 2: Key differences between the two approaches

6 Scalability Considerations

Both approaches were implemented using Apache Spark, ensuring distributed computation and the ability to scale to large datasets. The use of approximate similarity joins via Locality-Sensitive Hashing (LSH) allows the algorithms to find similar pairs without the need for expensive pairwise comparisons.

While the MinHash approach was faster, the TF-IDF method with cosine approximation was more flexible, though tuning the threshold for meaningful similarity and time considerations remained a challenge.

7 Experimental Results

7.1 Evaluation Criteria

We analyzed the top pairs returned by both approaches. The focus was on qualitative interpretability rather than quantitative accuracy, since no ground truth similarity labels were available.

7.2 Interpretation of MinHashLSH Results

Using the first approach (MinHashLSH), we identified several pairs of book reviews that share substantial textual overlap. The MinHashLSH method operates on the principle of estimating Jaccard similarity between sets of bigrams (two-word combinations), making it especially sensitive to direct textual matches and phrasing patterns. The results from the first approach using Jaccard distance reveal review pairs that share a significant amount of token overlap, indicating either partial reuse of content or very similar narrative structure. The closest match involves two reviews for High Fit, Low Fat Vegetarian, where the only differences are minor phrasings and punctuation. With a Jaccard distance of approximately 0.085, this pair reflects nearly identical content, reinforcing Jaccard's strength in detecting literal overlaps.
In the second pair of reviews for "The Biggest Loser Cookbook", there is considerable overlap in sentence structure, mentions of recipes (such as "Open Faced Burritos" and "Picante Chicken"), and common phrases (including "picky eater," "plain Jane," and "adapted from the TV show"). This similarity is reflected in a Jaccard distance of approximately 0.398, which indicates a strong content overlap, despite some variations in phrasing and sentence structure.
In the third pair, despite the books being different volumes from the same author (The Frugal Gourmet series), the reviews contain similar passages, especially around Jeff Smith's background and legacy. Both reviews share extensive content, such as "Jeff Smith entertained us for years on his PBS program 'The Frugal Gourmet'," and "Not only did he teach us many savory dishes, he also educated us." These recurring phrases and shared emotional tone account for the measurable similarity, despite referring to different titles. This similarity yields a moderate Jaccard distance (0.42), highlighting how the metric can detect repeated phrasing even within longer texts.
Overall, this approach excels at identifying duplicate or near-duplicate reviews, as well as those with substantial reused phrasing or common narrative elements. However, its reliance on word-level bigrams means that rephrased or paraphrased reviews with similar intent but different wording may be under-detected. Detected near-duplicate reviews effectively when bigram shingles captured overlapping phrases. However, increasing the n-gram size (e.g., trigrams or 4-grams) drastically reduced recall — only one pair was found under the 0.8 threshold.

Table 3: Top 3 Most Similar Review Pairs Based on Jaccard Distance

Title A	Title B	Review A	Review B	Distance
High Fit, Low Fat Vegetarian	High Fit, Low Fat Vegetarian	I own both the High Fit - Low Fat cookbooks (regular and vegetarian). The books are good for both...	I own both the High Fit - Low Fat cookbooks (regular and vegetarian). The books are good for both...	0.085
The Biggest Loser Cookbook: More Than 125 Healthy, Delicious Recipes Adapted from NBC's Hit Show	The Biggest Loser Cookbook: More Than 125 Healthy, Delicious Recipes Adapted from NBC's Hit Show	I am an incredibly picky eater and usually if I get 1 or 2 recipes out of a cookbook that I buy I'm pretty lucky. I'm pretty "plain jane" when it comes to food. The Biggest Loser Cookbook is...	This is a fun book that can help you eat healthier with a lot of good receipies.This review is from: The Biggest Loser Cookbook: More Than 125 Healthy, Delicious Recipes Adapted the...	0.398
The Frugal Gourmet Cooks Three Ancient Cuisines: China * Greece * Rome	The Frugal Gourmet on Our Immigrant Ancestors	This is an excellent cook book and this is the handy PB edition! It's full of great recipes and stories by a very talented cook and writer. This one focuses on 3 major influences in the culinary world. Jeff Smith...	My title blurb is a funny quote I remembered, Jeff Smith spoke on his entertaining PBS show. Before 'The Food Network' we had the witty and talented 'Frugal Gourmet'. This book deals with some simplistic...	0.424

7.3 Interpretation of BucketedRandomProjectionLSH

7.3.1 Raw Similar Review Pairs Based on Cosine Distance

The second approach evaluated similarity between user reviews by computing pairwise distances between their vector representations. Yet again, the pair with the lowest distance with near-duplicate text was for the book mentioned above, "High Fit, Low Fat Vegetarian". Being written by "different" users, this suggests potential duplication of content across user accounts. Even when book titles differ, as seen in the pair comparing Eating Well Through Cancer and Stalking the Wild Asparagus, the user feedback focuses on nearly identical delivery experiences, emphasizing timely shipping and good condition. The third pair, despite referencing different cookbooks, also centers around high praise using similarly structured language, suggesting that users may be echoing common expressions of satisfaction.

Table 4: Top 3 Most Similar Review Pairs Based on Cosine Distance (All)

Title A	**Title B**	**Review A**	**Review B**	**Distance**
High Fit, Low Fat Vegetarian	High Fit, Low Fat Vegetarian	I own both the High Fit - Low Fat cookbooks (regular and vegetarian). The books are good for...	I own both the High Fit - Low Fat cookbooks (regular and vegetarian). The books are good for...	2.00
Eating Well Through Cancer: Easy Recipes & Recommendations During and After Treatment	Stalking the Wild Asparagus	The book was in excellent condition, it came in a timely fashion. No problems.	Everything came in timely fashion and in excellant condition.Thank you	2.45
1000 Vegetarian Recipes From Around the World	Good Housekeeping Illustrated Cookbook	Whether you are vegetarian or not, this is the best cookbook I have ever owned.	this is best cookbook i have ever had. mine was falling apart. thank you	2.65

7.3.2 Filtered Similar Review Pairs (Distance > 2.5)

Title A	**Title B**	**Shared/Semantically Similar Phrases**
1000 Vegetarian Recipes From Around the World	Good Housekeeping Illustrated Cookbook	"this is the best cookbook I have ever owned", "this is best cookbook I have ever had"
How to Dry Foods	The New Spanish Table	"I look forward to trying...", "this book looks great", "I am enjoying reading this book"
Betty Crocker's Bisquick Cookbook	The Biggest Loser Cookbook	"the recipes are easy", "great book", "easy to read and follow"
The Biggest Loser Cookbook	The Biggest Loser Cookbook	"easy to follow", "different kinds of recipes", "I recommend it"
I'm Just Here for More Food	Simple French Food	"I enjoy his writing", "method for cooking", "I enjoyed it as I went"

Table 5: Top 5 similar review pairs (cosine distance) with shared/similar phrases

To further explore the accuracy of the second approach, we filtered out review pairs with a distance greater than 2.5 to exclude nearly identical reviews. The resulting review pairs exhibit strong semantic and lexical overlap, despite being associated with different books and users.

8 Discussion and Conclusion

The comparison between MinHashLSH with word-level bigrams and TF-IDF combined with BucketedRandomProjectionLSH (BRP-LSH) revealed key trade-offs in terms of precision, interpretability, and computational efficiency.

MinHashLSH, leveraging Jaccard similarity over word-level bigrams, demonstrated strengths in identifying reviews with overlapping vocabulary patterns. This approach is particularly effective in detecting near-duplicate texts or syntactically similar content. However, it lacks sensitivity to semantic differences, as it treats all tokens equally without considering their relative importance.

On the other hand, the TF-IDF + BRP-LSH approach incorporated term frequency and inverse document frequency, capturing the uniqueness and informativeness of words in each review. This made it more capable of identifying semantically similar reviews even when the exact wording differed. However, this method is more computationally intensive, especially due to the additional IDF weighting and the use of higher-dimensional vector representations. In practice, on a dataset of more than 11,000 reviews, the MinHashLSH approach completed in around **15-30 seconds**, whereas the TF-IDF + BRP-LSH pipeline took roughly **10-20 minutes**. This substantial difference highlights the increased overhead of computing TF-IDF vectors and performing LSH in a higher-dimensional space. That said, these runtimes are machine-dependent and can vary based on system specifications and Spark configuration. Nevertheless, this project demonstrates that scalable similarity detection can be achieved in Spark, and the choice of algorithm should reflect the nature of the data and the kind of similarity we want to capture.

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.