Collaborative Filtering based Book Recommender system

Here's a summary of the steps involved in creating this collaborative filtering-based book recommendation system, emphasizing the algorithmic aspects:

1. Data Filtering:
   - Selected users who have rated more than 200 books (active users).
   - Filtered for books that have been rated at least 50 times (popular books).
   - Created a subset of ratings data with these active users and popular books.

2. Data Transformation:
   - Created a pivot table with books as rows, users as columns, and ratings as values.
   - Filled missing values with 0, representing unrated books.

3. Similarity Calculation:
   - Used cosine similarity to compute the similarity between books based on their rating patterns.
   - This creates a square matrix where each element represents the similarity between two books.

4. Recommendation Algorithm:
   - For a given book:
     a. Find its index in the pivot table.
     b. Retrieve its similarity scores with all other books.
     c. Sort these scores to find the most similar books.
     d. Select the top 4 most similar books (excluding the input book itself).

5. Result Preparation:
   - For each recommended book, retrieve its title, author, and image URL.
   - Return this information as the recommendation output.

The core algorithm here is based on item-item collaborative filtering using cosine similarity. The key steps are:

1. Representing each book as a vector of user ratings.
2. Using cosine similarity to measure how similar these vectors are, which indicates how similar the books are in terms of user preferences.
3. Recommending books that are most similar to the input book, based on this similarity measure.

This approach leverages the collective behavior of users to find patterns in book preferences. It can provide personalized recommendations based on the rating patterns of similar items, even for books that a user hasn't rated yet. However, it may struggle with new or rarely-rated books (the cold start

problem) and can be computationally intensive for large datasets.