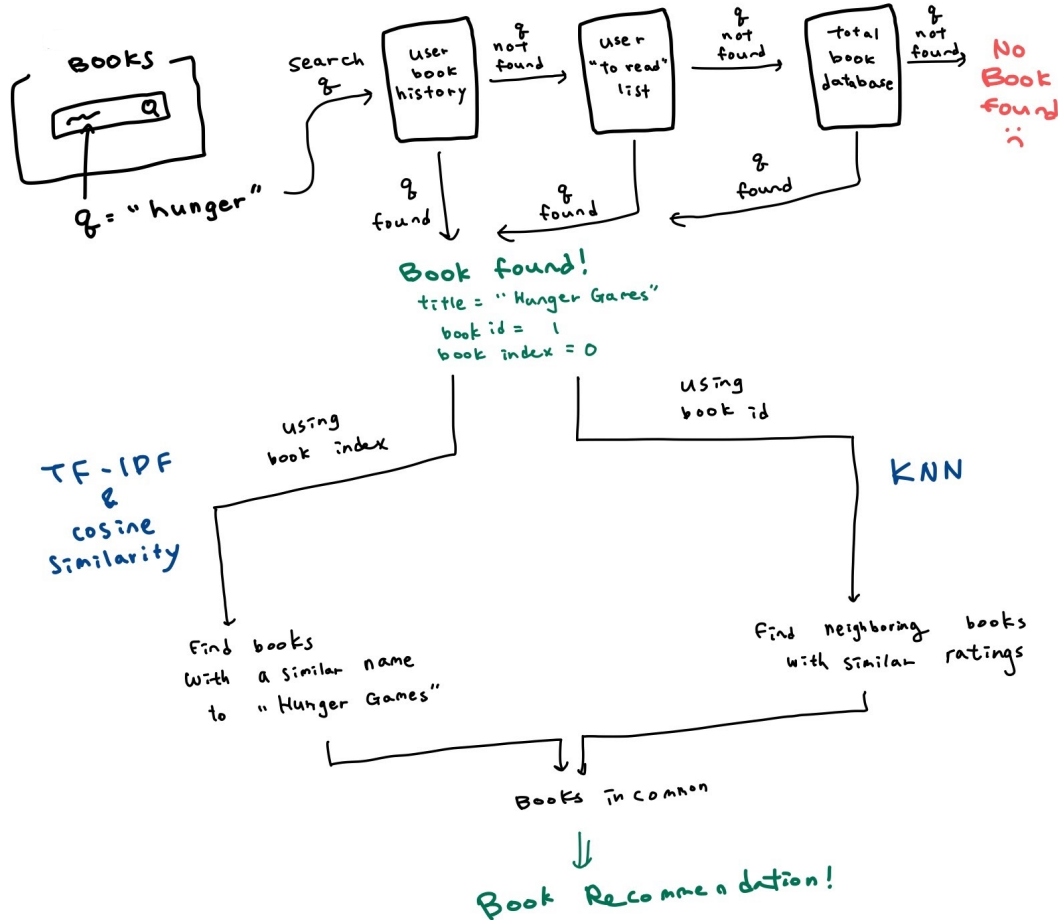**What is your problem formulation?**

What are the goals and non-goals of your model and system?
- Goals:
  - Implement both an information retrieval algorithm and an item-based collaborative filtering algorithm to provide recommendations that match the query as well as take the users' rating into account.
- Non-goals:
  - Allowing queries that are not written in English.
  - Making a recommendation list even if there are no books that match the query.

What features did you consider?
- Filter out tags that are not proper words.
- Filter out books that are not written in the same language to the input query.
- Do not recommend books that the user has already read.
- Do not recommend books with bad ratings ("IMDb" rating <= 3.2).
- Add more weights to the score of the book if it is in the user's "to read" list. Therefore the book can be prioritized for the personalized recommendation.

What model did you use and why?
- I used TF-IDF, Cosine Similarity, and KNN item-based collaborative filtering.
- TF-IDF and Cosine Similarity were used to find books that have a title, tag name, or author names that are associated with/ similar to the user's query. TF-IDF computes a weight to each word which reflects the importance of the word in the corpus.
- In addition, KNN item-based collaborative filtering was also used. Since the KNN approach relies on the books' similarities in terms of their ratings for the formation of a neighborhood of nearest books, unlike TF-IDF, we can leverage the users' rating information to provide a more reliable recommendation system.

What was your evaluation method for this? Why did you select this method? Please include details on how you evaluated your model.
- I used RMSE for KNN evaluation because it is measured in the same unit as the target variable so it is directly interpretable.
- For evaluating the book recommendation score, the accuracy score is the sum of the "IMDb" weighted rating and Cosine Similarity score (%). We chose this scoring method because, in this way, the accuracy score can reflect two important factors: 1. Is the book information (title, author names, tag names) similar to the query? 2. Does the book have a good rating?

What features would you like to add to the model in the future if you had more time?
- I would like to combine the current system with a KNN user-based collaborative filtering mode to find users that have similar book tastes. In this way, we would be able to find books to recommend even if there are no books that match the query.

What other things would you want to try before deploying this model in production?
- I would explore other methods for the information retrieval component of this system, such as a word2vec model to better find query-matching books.