Project Report

Overview:

This code takes a csv file of 10000 book ratings from goodreads that includes book_id, user_id, and rating. The output is the representatives of 5 book clusters, which is the book closest to the centroid of the cluster. It also outputs 5 recommendations for a user based on how they rated books that they have previously read. For the data, I used kaggle and found a dataset called ratings.csv from goodreads, which is a site that allows users to track what books they have read and review them. The dataset ratings.csv is downloadable from github at the link below.
Dataset: https://github.com/zygmuntz/goodbooks-10k/blob/master/ratings.csv

Modules:

load_data.rs: This module loads a csv file and transforms it into a matrix to be used. The rating struct stores individual ratings which each contain book_id, the user_id, and the rating that a user gave a book. The load_ratings function reads the csv file and parses each line into a Rating struct. It returns a Vec<Rating> with all the ratings from the CSV file. The build_rating_matrix function creates a 2D matrix where the rows represent users and the columns represent books. If a user has not rated a book, the matrix entry is 0.0. The function returns a matrix of shape [num_users, num_books].

similarity.rs: This module calculates the similarity between books using cosine similarity. The cosine_similarity calculates the cosine similarity between two vectors named v1 and v2. The function finds the products of two vectors as well as the norm of each of the vectors. It then returns the product of the two vectors divided by the product of the norm of the two vectors. The compute_similarity function calculates a similarity matrix for books using the cosine similarity previously calculated. This function returns a square matrix where each element in the matrix represents the similarity between two books. There is also a test in this module which checks the similarity matrix.

clusters.rs: This module implements k-means clustering on the books. The k_means_clustering function implements the k-means clustering algorithm. It assigns centroids and then for each point, finds the centroid that the point is closest to. Then, after all the points are assigned to centroids, it recalculates the new cluster center. This function returns a vector of cluster labels

where each point is assigned to a cluster. These steps are then repeated for a maximum number of iterations which is given by max_iterations or until the centroids stabilize. This module has a test that checks the k_means_clustering function.

cluster_reps.rs: This module selects the representative book for each cluster. This is the book that is closest to the centroid and therefore is the best representative of the cluster. In the select_rep function, for each cluster, the closest book to the centroid of the cluster is found by finding the minimum distance of each point to the centroid. This function returns a vector of book ids that each represent a cluster.

book_recs.rs: This module provides recommendations for users based on their previous ratings using the similarity matrix and clustering The recommend_books function recommends books to users based on the ratings and the ratings of similar users from the same cluster. It starts calculating the unweighted score of the books that are not rated by the user and then calculates the score based on the similarity of unrated books versus rated books. It then uses these scores and recommends the books associated with the highest scores. The function returns a list of recommended book IDs for the user.

main.rs: This module first loads user ratings from the ratings.csv CSV file. Then it converts the ratings into a 2D matrix with users represented by rows and books represented by columns. It then computes the similarity between the book. Then it performs k-means clustering on the book to group them together with k = 5. Then it generates 5 book recommendations for a specific user. It also generates the representatives of the clusters as well. Then it prints out the book representatives and the recommended books.

Output:

The output of the program should look similar to this:

Book Reps for each cluster: [8, 1, 2, 7, 4]

Recommended Books: [258, 4081, 260, 9296, 2318]

Based on the user_id that is chosen and the amount of clusters created, this output can change.