Technical Specification:

1. Overview: To build a recommender system.

Technology stack: Springboot, Rest, Apache Mahout, Factory design pattern

Github repository:

<https://github.com/ilakkmanoharan/bookrecommender>

2. How to run this app?

mvn clean package

./mvnw spring-boot:run

3. API Endpoints:

3. 1 Get recommendations for a user

GET recommend/{userid}

Parameters:

recommendertype

similarityType

neighborhoodType

Default options:

recommendertype=UserBasedRecommendation

similarityType=Spearman

neighborhoodType=NearestN

3.2 Get results for an evaluator

GET evaluate/

Parameters:

recommendertype

similarityType

neighborhoodType

Default options:

recommendertype=UserBasedRecommendation

similarityType=Spearman

neighborhoodType=NearestN

3.3 Few example URLs:

GET recommend/792?recommendertype=UserBasedRecommendation

GET recommend/792?recommendertype=ItemBasedRecommendation

GET recommend/792?recommendertype=ItemBasedRecommendation&similarityType=Spearman&neighborhoodType=NearestN

GET evaluate/?evaltype=absoluteDifferenceEvaluation&recommendertype=UserBasedRecommendation

GET evaluate/?evaltype=rmsEvaluation&recommendertype=UserBasedRecommendation

GET evaluate/?evaltype=irStatsEvaluation&recommendertype=UserBasedRecommendation

GET evaluate/?evaltype=absoluteDifferenceEvaluation&recommendertype=ItemBasedRecommendation

GET evaluate/?evaltype=rmsEvaluation&recommendertype=ItemBasedRecommendation

GET evaluate/?evaltype=irStatsEvaluation&recommendertype=ItemBasedRecommendation

4. How recommendations are made and what is covered?

The recommendations are made using the collaborative filtering technique:

Collaborative filtering uses user's historical preference on a set of items. In this assignment I am using explicit rating data from the userrating.csv file.

I have implemented user based CF and item based CF. In the user based CF, the nearest neighbourhood algorithm is used. Here the similarities are calculated between target user and all other users, top X similar users are selected and taken the weighted average of ratings from these X users with similarities as weights.

Different ways to calculate user similarity implemented in this project are: Spearman (default option), Pearson, Cosine and Euclidean.

Basically, the idea is to find the most similar users to your target user (nearest neighbors) and weight their ratings of an item as the prediction of the rating of this item for target user.

For Item based CF, two items are considered as similar if they received simillar ratings from a same user. The weighted average of ratings on most X similar items from a target user is calculated and the prediction is made for a target user on an item.

Since we have only two ratings scale - 0 or 1 (for like and dislike respectively) in this dataset, we donot have to worry about people having different baselines when giving rating.

5. Out of Scope and potential future developments:

It is planned to load the books.csv dataset into the mongodb database so that it is easier to map book name and other details by book id (or ASIN). But this is not covered in this MVP.

6. Observation:

The relation between Spearman (S) and Pearson (P) correlations gives information on the data. Spearman depicts monotonic relationships while Pearson depicts linear relationships.

It is found that we have S > P which indicates that the movie dataset leans towards correlation that is monotonic but not linear