International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-11, Issue-1, October 2021

Apache Mahout based Book Recommendation System

Abhishek Verma, V Nallarasan

Abstract*: E-Commerce websites plays an important role in an individual's life as it serves as the medium for online shopping with a huge audience. With the commencement of the pandemic due to novel coronavirus, the involvement of E-Commerce websites for shopping has drastically increased or more precisely it remains as the only medium to shop. With the increasing demand for online shopping on E-Commerce websites, the role of the Recommendation System has also become vital as it accomplishes the goal to make Personalized Recommendation for users. In this paper, we set out Apache Mahout-based Book Recommendation System to help recommend books to users. With this paper, we have described our project that recommends books to users on the basis of the user's prior experience of purchase. The platform utilizing this recommendation system is developed using Spring Framework as a part of our project. The dataset used in the process is taken from Kaggle. Dataset has ratings for various books given by users. As a part of the User-based Collaborative Filtering recommendation technique, Euclidean Distance Similarity is used as a similarity measure along with Nearest N User Neighborhood and Generic User-Based Recommender to give quality results as compared to the existing system. To get the best quality recommendation we have obtained an evaluation score of 0.5 for Euclidean Distance Similarity.

***Keywords:** Apache Mahout, Book Recommendation, Collaborative Filtering, Machine Learning, Spring Framework, Web Application.*

I. INTRODUCTION

With the commencement of the pandemic due to novel coronavirus, the involvement of E-Commerce websites for shopping has drastically increased or more precisely it remains as the only medium to shop. With such an increase in the demand for online shopping platforms, it is important that users get the best quality recommendations based on the purchases made in past. Our project uses Apache Mahout, a Machine Learning framework to implement Collaborative Filtering based recommendation system to recommend books to users on an online book shopping platform developed using Spring Framework. The focus of the project is to help customers get the best quality recommendations and avail the best offerings from an E-Commerce platform when the online mode is the most suited way for shopping. The recommendation system is of immense value in a current situation not only restricted to books but also for a vast variety of products including various essentials required during the pandemic.

II. LITERATURE REVIEW

The previous researcher's Machine Learning techniques and approaches for recommending items have been quite successful. Saikat Bagchi at IIT Kharagpur [1] has analyzed and compared various similarity measures which is an important aspect to make recommendations using Collaborative Filtering. As a result of the study conducted it is concluded that Euclidean Distance Similarity performs very well as compared to other similarity measures which are being used in our project. [2] Dilek Tapucu, Seda Kasap, Fatih Tekbacak have shown combined solution results using various similarity measures. They have described that Pearson Correlation which is user-based CF algorithm has a better performance. They have also proved that combined user and item-based CF algorithms can perform better in some scenarios. [3] Johnpaul C I, Neetha Susan Thampi, Dr. Senthil Kumar Thangavel have concluded that Apache Mahout can handle a large amount of structured data which is being used in our project. [4] Ananya Agarwal, S. Srinivasan have used Pearson Correlation Similarity as a similarity measure in the Collaborative Filtering technique for building a Movie Recommendation System. [5] Abhilasha Sase, Kritika Varun, Sanyukta Rathod, Deepali Patil have proposed that a hybrid recommendation system is more accurate and efficient as it combines the features of various recommendation techniques.

In summary, there are many existing works around Collaborative Filtering Based Recommendation Systems and most of them use data being provided by the users on E-Commerce sites to recommend items to them. There are works existing around Book Recommendation Systems as well. Due to the sudden increase in demand of online shopping, it is vital that customers get the best quality recommendations and avail best offerings from these sites for the books or any other items that they purchase.

Manuscript received on January 10, 2021.
Revised Manuscript received on January 20, 2021.
Manuscript published on January 30, 2021.

* Correspondence Author
Abhishek Verma*, Department of Information Technology, SRM Institute of Science and Technology, Chennai (Tamil Nadu), India. Email: abhishek.verma4607@gmail.com
V Nallarasan, Assistant Professor, Department of Information Technology, SRM Institute of Science and Technology, Chennai (Tamil Nadu), India. Email: nallarav@srmist.edu.in

III. DATASET DESCRIPTION

The dataset for Book Recommendation was taken from Kaggle, a repository for data and scientific papers related to diverse fields and can be accessed online.

This dataset consists of three columns: the first column as User Id, the second column as Book Id, and the third column as Book Rating. Every user will have a unique id and can give ratings ranging from '1' to '5' with the higher value being the best rating for every book having a unique book id. The dataset has 981756 entries with users giving ratings to various books ranging from '1' to '5'.

IV. METHODOLOGY

The methodology used in our project is based on the evaluation of various similarity measures present in Apache Mahout followed by the comparison of evaluation scores (MAE) to select the best similarity measure and then use it to give quality recommendations to users.

A. Similarity Measure and Abbreviated Form

The below mentioned similarity measures were taken into consideration for evaluation in our project.

- Euclidian Distance Similarity (EDS)
- Uncentered Cosine Similarity (UCS)
- City Block Similarity (CBS)
- Log-Likelihood Similarity (LLS)
- Tanimoto Correlation Similarity (TCS)
- Spearman Correlation Similarity (SCS)
- Pearson Correlation Similarity (PCS)

B. Evaluation of Recommender

Recommendation Systems try to predict the ratings for those items for which the user has not given any preference. The difference between the predicted rating and the actual rating explains the quality of estimated preference values.

Evaluation of Score: The real data is divided into training data and test data and then the recommendation system estimates the preferences for the test data so that it can be compared with the actual preferences to estimate the quality of recommendation. A score can be calculated for a recommender after evaluation which is equivalent to the average difference between the actual and the estimated preferences. A lower score value shows that the estimated preference is near to the actual preference values.

C. Comparison of Similarity Measure

We have used Average Absolute Difference Recommender Evaluator to generate the evaluation score for each similarity measure with 10% data as testing data and 90% data as training data. The code showing the evaluation score of the similarity measure of the recommender being calculated is shown below in "Fig. 1" and "Fig. 2".

```
package com.recommendationengine;

import org.apache.mahout.cf.taste.common.TasteException;
import org.apache.mahout.cf.taste.eval.RecommenderBuilder;
import org.apache.mahout.cf.taste.impl.neighborhood.NearestNUserNeighborhood;
import org.apache.mahout.cf.taste.impl.recommender.GenericUserBasedRecommender;
import org.apache.mahout.cf.taste.impl.similarity.EuclideanDistanceSimilarity;
import org.apache.mahout.cf.taste.model.DataModel;
import org.apache.mahout.cf.taste.neighborhood.UserNeighborhood;
import org.apache.mahout.cf.taste.recommender.Recommender;
import org.apache.mahout.cf.taste.recommender.UserBasedRecommender;
import org.apache.mahout.cf.taste.similarity.UserSimilarity;

public class BookRecommender implements RecommenderBuilder{

    public Recommender buildRecommender(DataModel dataModel) throws TasteException {
        UserSimilarity similarity= new EuclideanDistanceSimilarity(dataModel);
        UserNeighborhood neighborhood=new NearestNUserNeighborhood(5, similarity, dataModel);
        UserBasedRecommender recommender = new GenericUserBasedRecommender(dataModel, neighborhood, similarity);
       return recommender;
    }
}
```

Fig. 1. Snapshot - Evaluation Score calculation of Similarity Measure

```
    System.out.println("Recommendations for customer " + userNames[0] + " are :");

    System.out.println("**********************************************************");

    System.out.println("BookId\ttitle\t\testimated preference");



    for (RecommendedItem recommendedItem : recommendations) {
       int bookId=(int)recommendedItem.getItemID();
       float estimatedPref = recommender.estimatePreference(1, bookId);
       System.out.println(bookId+" "+ books[bookId-1]+"\t"+estimatedPref);
    }

    System.out.println("**************************************************");
    long[] userIds=recommender.mostSimilarUserIDs(1, 5);
    System.out.println("Most similar users for "+ userNames[0] +" are");
    for (long id : userIds) {
       System.out.println(id+" "+userNames[(int)id-1]);
    }

    RecommenderEvaluator evaluator= new AverageAbsoluteDifferenceRecommenderEvaluator();

    RecommenderBuilder builder = new BookRecommender();

    double result=evaluator.evaluate(builder, null, model, 0.9, 1.0);

    System.out.println(result);

    }
```

Fig. 2. Snapshot - Evaluation Score calculation of Similarity Measure

The chart showing the evaluation scores of various similarity measures is shown in "Fig. 3".

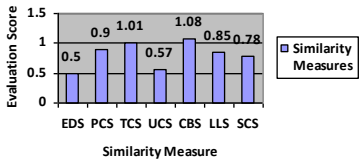

Fig. 3. Evaluation Scores of various Similarity Measures for Testing Data of 10% and Training Data of 90%

Based on the comparison made from our experiment we concluded that Euclidean Distance Similarity has the lowest score and so it provides the best quality recommendation.

V. MODULE DESCRIPTION

We have used the Euclidean Distance Similarity, Nearest N User Neighborhood, and Generic User-Based Recommender in our project.

A. Euclidean Distance Similarity

The Euclidean Distance Similarity assumes items as

dimensions and ratings as points along those dimensions, a distance is calculated using all items (dimensions) where both users have given ratings for that item. The coordinates for the points between which the Euclidean Distance must be calculated are shown in "Fig. 4" and the Euclidean Distance expression is shown in "Fig. 5" which is obtained by calculating the difference in rating i.e., position along each dimension, calculating the sum of squares of those differences and then taking the square root of it.

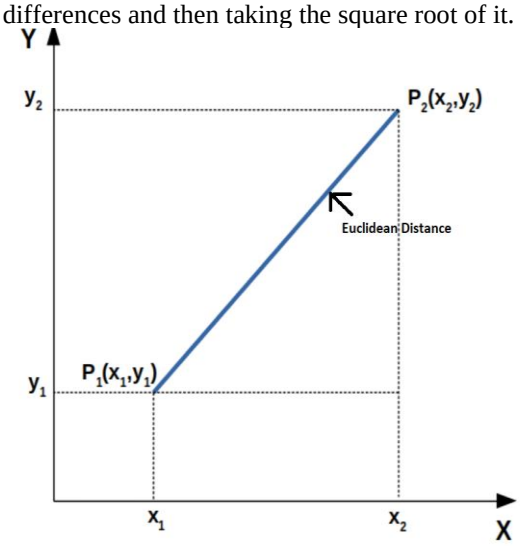

Fig. 4. Euclidean Distance

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

Fig. 5. Euclidean Distance Formula

Here n is the number of dimensions. The Euclidean Distance Similarity can be calculated by using the expression shown in "Fig. 6".

$$S = \frac{1}{(1 + d(x, y))}$$

Fig. 6. Euclidean Distance Similarity Formula

which results in a value between 0 and 1.

B. Nearest N User Neighborhood

The algorithm computes a neighborhood having the nearest N users to a given user based on the User Similarity value calculated. The first N users with the highest similarity value are considered as neighbors. Here, N is the neighborhood size.

Also, the minimum value of similarity is needed as the threshold for consideration of similarity value to compute the neighbors. Factors like sampling rate which is the percentage of users to consider when building the neighborhood are also taken into consideration. Diagrammatic representation for the same is shown in "Fig. 7".

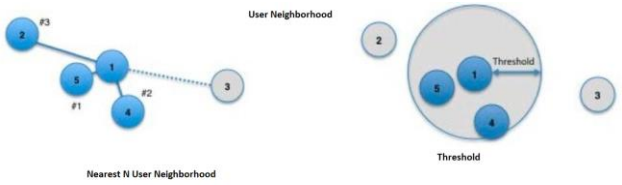

Fig. 7. Nearest N User Neighborhood

C. Generic User Based Recommender

A simple Generic User Based Recommender uses a given Data Model and User Neighborhood to produce recommendations.

The preference for a particular item is taken from the data model for all the users in the neighbor of the target user and similarly the similarity is also calculated between the target user and its neighbors. The total preference is calculated by multiplying the value of similarity calculated between the target user and each user in the neighbor with the preference of the respective users in the neighbor for a particular item and then summing it up as shown in "Fig. 8". The formula to calculate the total similarity is shown in "Fig. 9".

$$\text{Total Preference} = \sum_{i=1}^{i=n} \text{Similarity}(\text{theUserId}, \text{UserId}_i) * \text{Preference}(\text{UserId}_i, \text{ItemId})$$

Fig. 8. Total Preference Formula

theUserId = Target user for which the similarity needs to be compared and calculated.

UserId = The users in the neighbor of the target user.

Preference = Preference for a particular item by the users in the neighbor of the target user.

$$\text{Total Similarity} = \sum_{i=1}^{i=n} \text{Similarity}(\text{theUserId}, \text{UserId}_i)$$

Fig. 9. Total Similarity Formula

theUserId = Target user for which the similarity needs to be compared and calculated.

UserId = The users in the neighbor of the target user.

Thus, total similarity is the sum of similarities between each pair (of target user and other users) in the neighborhood. The estimated preference is obtained by using the formula shown in "Fig. 10".

$$\text{Estimate} = \frac{\text{Total Preference}}{\text{Total Similarity}}$$

Fig. 10. Estimated Preference Formula

In our project we have used the Euclidean

Distance Similarity, Nearest N User Neighborhood and Generic User Based Recommender on sample data being used in our application. A snapshot of the code is shown in "Fig. 11" and the book recommendations for a specific customer along with estimated preference and the most similar users as compared to that specific user is shown in "Fig. 12".

```
public static void main(String[] args)throws IOException, TasteException{

 model=new FileDataModel(new File(dataSet));
 similarity = new EuclideanDistanceSimilarity(model);
 neighborhood = new NearestNUserNeighborhood(NEIGHBORHOOD_SIZE, similarity, model);
 recommender = new GenericUserBasedRecommender(model, neighborhood, similarity);

 List<RecommendedItem> recommendations = recommender.recommend(1, 5);

 System.out.println("Recommendations for customer " + userNames[0] + " are :");

 System.out.println("*************************************************************");

 System.out.println("BookId\ttitle\t\testimated preference");

 for (RecommendedItem recommendedItem : recommendations) {
    int bookId=(int)recommendedItem.getItemID();
    float estimatedPref = recommender.estimatePreference(1, bookId);
    System.out.println(bookId+" "+ books[bookId-1]+"\t"+estimatedPref);
 }


 System.out.println("*************************************************");
 long[] userIds=recommender.mostSimilarUserIDs(1, 5);
 System.out.println("Most similar users for "+ userNames[0] +" are");
 for (long id : userIds) {
    System.out.println(id+" "+userNames[(int)id-1]);
 }

 }
```

Fig. 11. Snapshot - Euclidean Distance Similarity, Nearest N User Neighborhood and Generic User Based Recommender

```
Recommendations for customer Abhishek are :
*************************************************************
BookId  title           estimated preference
6 Romeo and Juliet      3.6591632
3 Pride and Prejudice   2.2316382
7 The Alchemist 2.2065778
*************************************************
Most similar users for Abhishek are
15 Aditya
5 Vikas
16 Rahul
20 Nani
7 Ravinder
```

Fig. 12. Recommended Books and Most Similar Users

D. Spring Framework

The web application is developed using Spring Framework and can recommend books based on the ratings given by the user in past after calculating the values based on the Euclidean Distance Similarity, Nearest N User Neighborhood, and Generic User-Based Recommender as explained above. The Online Shopping Platform provides features like adding books to the cart, removing an item from the cart, placing an order after viewing the total value, rating a specific book apart from recommendations with the data being persisted in a MySQL Database.

VI. SYSTEM ARCHITECTURE

The components involved in Apache Mahout User Based Recommender is shown below.

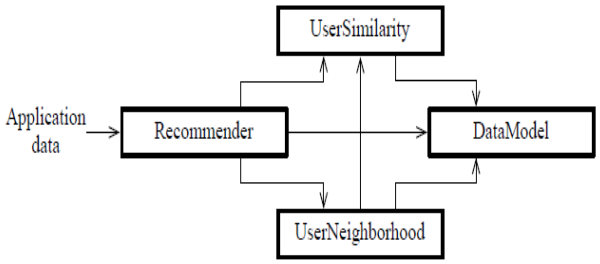

Fig. 13. Interaction between components in Apache Mahout User Based Recommender

VII. PERFORMANCE OF FINAL METHOD

As we are recommending books to users based on the User-based Collaborative Filtering Algorithm. The similarity measure plays a vital role in deciding the quality of recommendations being made to the user. Based on the experiment conducted we arrived at a conclusion that Euclidean Distance Similarity (EDS) has the lowest evaluation score of 0.5 as compared to other similarity measures and thus provides the best quality recommendation.

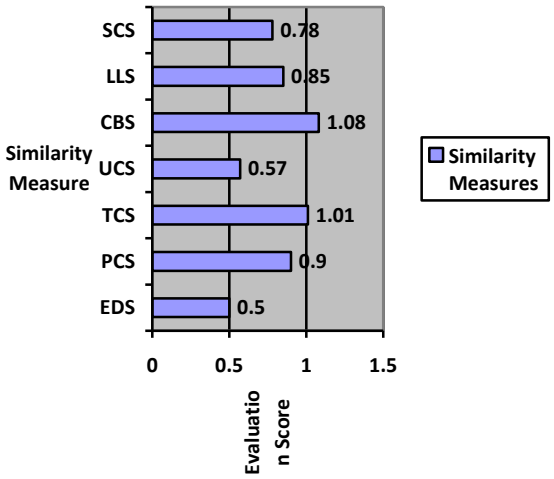

Fig. 14. Evaluation Score of various Similarity Measures calculated based on MAE

VIII. CONCLUSION AND FUTURE WORK

In our model, we are using Euclidian Distance Similarity as the similarity measure to produce best quality recommendations because it has a low evaluation score as shown in "Fig. 14". Perforce, we conclude that the Euclidean Distance Similarity has been found to be the most appropriate similarity measure for providing quality recommendations. We bring forward this substructure to assist users to shop online by getting quality recommendations. This same methodology can be used for recommending a variety of products to users. For our future work, we would like to look at how the proposed methodology works with datasets containing a variety of products including scenarios where all the products are not rated.

REFERENCES

1. Saikat Bagchi (2015) - "Performance and Quality Assessment of Similarity Measures in Collaborative Filtering Using Mahout": IIT Kharagpur, India.
2. Ananya Agarwal, S. Srinivasan (2020) - "Movie Recommendation System" IRJET: Department of Computer Science, Galgotias University, Uttar Pradesh.
3. Dilek Tapucu, Seda Kasap, Fatih Tekbacak (2012) – "Performance Comparison of Combined Collaborative Filtering Algorithm for Recommender Systems" IEEE: Department of Computer Engineering, Izmir Institute of Technology, Urla, Izmir, Turkey and Faculty of Engineering and Natural Sciences, Sabanci University, Orhanli, Tuzla, Istanbul, Turkey.
4. Taner Arsan, Efecan Koksal, Zeki Bozkus (2016) – "Comparison of Collaborative Filtering Algorithm with various Similarity Measures for Movie Recommendation" IJCSEA: Department of Computer Engineering, Kadir Has University, Istanbul, Turkey.
5. Abhilasha Sase, Kritika Varun, Sanyukta Rathod, Deepali Patil (2015) – "A Proposed Book Recommender System" IJARCCE: Sinhgad Institute of Science and Technology, Pune, India.
6. Tarun Bhatia, Upendra Chaurasia (2016) - "User-Based Collaborative Filtering Recommendation System using Apache Mahout", Gregoria Institute of Science and Technology.
7. Dr. Senthil Kumar Thangavel, Neetha Susan Thampi, Johnpaul C I (2013) – "Performance Analysis of Various Recommendation Algorithms Using Apache Hadoop and Mahout", IJSER.
8. S. Kanetkar, Akshay Nayak, S. Swamy, G. Bhatia (2014) - "Web-based personalized hybrid book recommendation system", ICAETR.
9. https://mahout.apache.org/
10. S. G. Walunj, K Sadafale, "An online recommendation system for e-commerce based on Apache Mahout framework", 2017 ACM SIGMIS International Conference on Computers and People Research, pp 153-158, 2013.
11. Z D Zhao, M. Shang," User-Based collaborative filtering recommendation algorithms on Hadoop", Proc. Of Third International Workshop on Knowledge Discovery and Data Mining, pp. 478-481, 2016.
12. A. V. Dev and A. Mohan," Recommender system for big data applications based on set similarity of the user preferences" 2016 International Conference of on Next Generation Intelligent Systems (ICNGIS), Kottayam. 2016, pp.1-6.

AUTHORS PROFILE

Abhishek Verma, B. Tech with specialization in Information Technology from SRM Institute of Science and Technology, Kattankulathur, 2020 graduate. Software Engineer. Core Interest: Machine Learning, Algorithm Development, Java, Web Application Development.

V Nallarasan, Assistant Professor (OG) at SRM Institute of Science & Technology, Kattankulathur. M.E with specialization in Computer Science form Karpagam University, 2012. Research Interest: Bigdata, Machine Learning.