**BIL 476 Data Mining Project Interim Report**

**Recommendation System With Collaborative Filtering**

Kasım Alp TOKER Ayça GÜRLEYİK Orhan ONAR

131101058 151101032 131101063

**ABSTRACT**

In this project, our goal is to enhance a Book Recommendation System with using Collaborative Filtering.In this way, we will equip the local website to demo this system which consist recommendation system with Collaborative Filtering.

**1. INTRODUCTION**

At the present time, people may use the internet to access the alternative of their favourite objects quite easily. However, in the acquiring of alternatives, people generally choose alternatives which are popular. Because of this choice, the person misses a lot of alternatives that fit him/her more. For instance, the person took the book which is not genre of that person likes because that person saw it on the list of popular boks that day and, he could not read that book. Even when people start reading a new genre, they ask the question of where to start. They find what they want to read after they read many books which are not good choices for that person. We intend to integrate a Recommendation System which is created using Collaborative Filtering to shorten the duration of the “long-running adventure”.

**2. RECOMMENDATION SYSTEMS**

[1] Recommendation System or otherwise known as Recommendation Engines is a subclass of the Information Filtering Systems. It is often used in advertising (commercial) applications. Major companies such as Netflix, Youtube and Spotify use the Recommendation System. There are many Recommendation System approaches. Some of these approaches are Collaborative Filtering, Content-Based Filtering, Multi-Criteria Systems.

In this project, we will use Collaborative Filtering which is an approach of the Recommendation System.

**2.1 COLLABORATIVE FILTERING**

[2] Collaborative Filtering is divided into two approaches. These approaches are Narrow and General.

The Narrow Approach which is the newest attempts to predict a user’s interest by taking the idea of many users.

In general, Collaborative Filtering is a filtering process. Informations and patterns are used during this process.

The Collaborative Filtering assumption is that if person A agreed with Person B on a topic in the past, person A is close to agreement on a topic in the future according to a random person.

Collaborative Filtering basically consists of two phases;

* Scan for other users who agree with the user.
* Create an estimate for the user by using the information of the users who agree.

**3. IMPLEMENTATION**

Our implementation will mainly be on the website. We generate a site for trial rather than a real site because our goal is to have more Collaborative Filtering implementation. Technologies and descriptions are located in the 4. Technologies section. Implementation images are located in the 5. Appendix section.

**3.1 IMPLEMENTATION ABSTRACT**

We do not plan to use any “Register Section”, user information is added to the database manually, then user starts adding the books which are read by user to user’s profile. Important case in our project is that when adding one’s book “Genre” means adding the type of the book. We aimed to use the “Genre” attribute in our project instead of the “Rating” used in common Collaborative Filtering.

We thought of using “Rating” to constitute a Recommendation System according to the “Genre” attribute to prevent people from missing out on books which are liked by them, as described in the abstract section. When the user clicks on any books that he/she has added after his/her books, the books of the closest type are listed and user can add the books from that list to his/her list. The user’s books are included in our Recommendation System is detailed in the section 3.2. Algorithm.

**3.2 ALGORITHM**

First of all, our algorithm expects user inputs. As mentioned in the 3.1 Implementation Abstrat section, the user must add the book “X” his/her profile then click on the book “X” to see the books which are closest to the book “X” in the “Genre”. The information of book “X” is taken whent user clicks into this book. (Book name,author,year,type etc.). Then, to make a search faster, instead of iterating through the whole books dataset, we narrow the dataset to the type of book “X”. When sorting the books closest to the book “X”, if the user selects the by author option, searches for the type of the “X” book + the author of the “X” book will be the first option.The next books will be the ones that do not correlate. In the same way, the person can perform this search according to the year of the book. For example, if the user adds the book “Y” from this list to his/her list, the name of this list will be “Read List”. When the user clicks on the “X” book , the “Y” book will appear on the list until the “Y” book enters the “Read Books” list.

**3.3 DATASET**

Collaborative Filtering works more effectively in fairly large datasets. Accordingly, we first wanted to use the dataset of the [3] Good Books 10K project in the [4] Kaggle, but we had some problems with the “Tags”, that is the “Genre” in this dataset Because the Good Books 10K project is a Recommendation System created according to “Rating”, the dataset has not been an appropriate set for us. The different datasets that we found were too large, the DBMS that we used were insufficient. So, we decided to take care of other issues instead of dataset which includes “Genre” until the interim report is finished. For now, we will work on a trial dataset.

**4. TECHNOLOGIES**

For this project PHP, HTML, SQL, CSS As the framework [5]CodeIgniter (Model View Controller structure similar to framework RubyOnRails)

For Database Management System (DBMS) [6]HeidiSQL we are planning to use. To be able to work in local, we will use [7]Xampp.

**5. APPENDIX**

**6. REFERENCES**

[1] : Reccomendation Systems <https://en.wikipedia.org/wiki/Recommender_system>

Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1-35

[2] : Collaborative Filtering <https://en.wikipedia.org/wiki/Collaborative_filtering>

[3] : Kaggle [https://www.kaggle.com](https://www.kaggle.com/)

[4] : Goodbooks10K <https://www.kaggle.com/zygmunt/goodbooks-10k>

[5] : Codeigniter <https://www.codeigniter.com/>

[6] : HeidiSQL <https://www.heidisql.com/>

[7] : Xampp <https://www.apachefriends.org/tr/index.html>