**Movie Recommendation Web Application**

**Submitted By:**

Arjun Singh Chauhan (9918103063)

Heena Nankani (9918103102)

Kapil Sharma (9918103104)

Under the supervision of: Mrs. Ambalika Sarkar



**Department of CSE/IT**

**Jaypee Institute of Information Technology University, Noida**

# **October 2021**

Submitted in partial fulfillment of the Degree of Bachelor of Technology

in

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

**TABLE OF CONTENTS**

**Page No.**

**Chapter-1 Introduction**

|  |  |  |
| --- | --- | --- |
| 1.1 | General Introduction | 3 |
| 1.2 | Problem Statement | 3 |
| 1.3 | Significance of the problem | 3 |
| 1.4 | Empirical Study | 4-6 |
| 1.5 | Brief Description of the Solution Approach | 6 |
| 1.6 | Comparison of existing approaches to the problem framed | 6 |

**Chapter-2 Literature Survey**

|  |  |  |
| --- | --- | --- |
| 2.1 | Summary pf the papers studied | 7-9 |
| 2.2 | Integrated summary of the literature studied | **9** |

**Chapter-3 Requirement Analysis and Solution Approach**

|  |  |  |
| --- | --- | --- |
| 3.1 | Solution Approach |  |

**Chapter-4 Modeling and Implementation Details**

|  |  |  |
| --- | --- | --- |
| 4.1 | Implementation details and issues |  |

**Introduction**

Recommender System (RS) has emerged as a major research interest that aims to help users to find items online by providing suggestions that closely match their interests. Recommendation systems are very crucial for a good business on a website. A robustly implemented recommendation system can provide a better user experience to the user and allow him to make better choices for his transactions.

**Problem Statement**

In this project we aimed to create a movie recommendation web application which would allow users to make informed decisions to decide the movie choice depending on their taste. We would be creating multiple models which would collectively be providing a better user experience.

**Significance of the Problem**

Recommender system has the ability to predict whether a particular user would prefer an item or not based on the user's profile. Recommender systems are beneficial to both service providers and users. They reduce transaction costs of finding and selecting items in an online shopping environment

**Proposed Solution**

In this project we have aimed to create a complete working prototype of a movie recommending website which apart from just displaying different movies would also consider the choices of each logged in user and provide the recommendation on its own.

To make the website more interactive we have tried to implement a dynamic model which will try to keep learning as the new data would be collected for the new users and new data for the existing users.

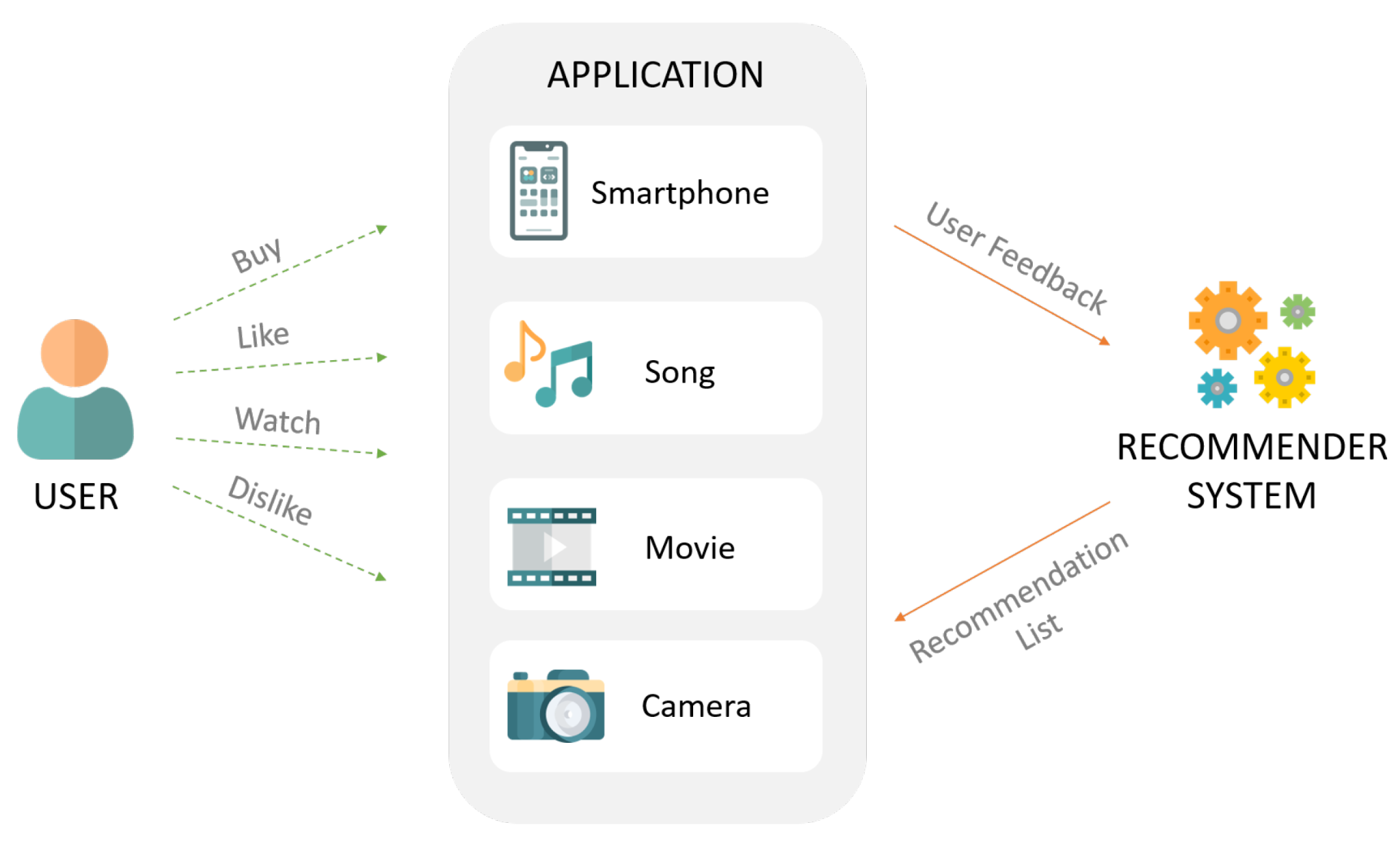
**Empirical Study**

Currently there are multiple models that have been trained on the movie lens dataset but not many applications that are able to harness the strengths of all these models to provide a robust solution.

What we are trying to create is not just a trained model but a complete model which would be backed by multiple models which would collectively provide a much more meaningful solution.

**What is a recommender system?**

A recommender system is a simple algorithm whose aim is to provide the most relevant information to a user by discovering patterns in a dataset. The algorithm rates the items and shows the user the items that they would rate highly. An example of recommendation in action is when you visit Amazon and you notice that some items are being recommended to you or when Netflix recommends certain movies to you. They are also used by Music streaming applications such as Spotify and Deezer to recommend music that you might like.

****

**Different types of recommendations engines:**

The most common recommendation systems are content-based and collaborative filtering recommender systems. In collaborative filtering, the behavior of a group of users is used to make recommendations to other users. The recommendation is based on the preferences of the other user based on the fact that their friend liked the movie. There are two types of collaborative models: **Memory-based** methods and **Model-based** methods. The advantage of memory-based techniques is that they are simple to implement and the resulting recommendations are often easy to explain. They are divided into two:

* **User-based collaborative filtering:** In this model, products are recommended to a user based on the fact that the products have been liked by users similar to the user. For example, if Derrick and Dennis like the same movies and a new movie comes out that Derrick Likes, then we can recommend that movie to Dennis because Derrick and Dennis seem to like the same movies.
* **Item-based collaborative filtering: ·** These systems identify similar items based on users’ previous ratings. For example, if users A, B, and C gave a 5-star rating to books X and Y then when a user D buys book Y they also get a recommendation to purchase book X because the system identifies book X and Y as similar based on the ratings of users A, B, and C.

Model-based methods are based on Matrix Factorization and are better at dealing with sparsity. They are developed using data mining, machine learning algorithms to predict users’ rating of unrated items. In this approach techniques such as dimensionality reduction are used to improve accuracy. Examples of such model-based methods include Decision trees, Rule-based Model, Bayesian Model, and latent factor models.

**Content-based systems** use metadata such as genre, producer, actor, musician to recommend items, say movies or music. Such a recommendation would be for instance recommending Infinity War that featured Vin Diesel because someone watched and liked The Fate of the Furious. Similarly, you can get music recommendations from certain artists because you liked their music. Content-based systems are based on the idea that if you liked a certain item you are most likely to like something that is similar to it.

**Datasets to use for building recommender systems:**

In this project, we are going to use the Movie Lens Data Set. This dataset was put together by the Group lens research group at the University of Minnesota. It contains 1, 10, and 20 million ratings. Movie lens also has a website where you can sign up, contribute reviews and get movie recommendations.

**Brief Discussion of Approach**

* We aim to create a complete web application that has implementation of a machine learning-based recommendation system
* We would be using both content-based filtering and collaborative approach for the recommendation system, we may use a Hybrid model of both the above-given models. The former will be used for new users who would be recommended the movies which are most commonly viewed by users, and the latter will be used to recommend movies on the likeness of the user based on the movies he likes.
* We also aim to maintain a complete database for recording the reviews of users on various movies so that other users can easily view them before they consider a movie.
* Each user will be presented with a customized page that would be specific to the user and would be created dynamically based on his or her choices.
* We would be also deploying the project on AWS and probably consider putting it behind a load balancer to mimic a real use case.

**Comparison to the previous approaches:**

The previous approaches for this usually rely on one filtering model but this project showcases a hybrid of 2 models that will be used to recommend movies to the user also the training dataset is more optimized and more filtered so that filtering works flawlessly.

**Literature Survey**

**2.1 Summary of Papers studied**

**2.1.1**

One of the potent personalization technologies powering the adaptive web is collaborative filtering. Collaborative filtering (CF) is the process of filtering or evaluating items through the opinions of other people.CF technology brings together the opinions of large interconnected communities on the web, supporting the filtering of substantial quantities of data. Collaborative Filtering System Functionality There are also broad abstract families of tasks that CF systems support. It is no accident that this system functionality is related to the user tasks of the previous section. Ideally, the system would support all user tasks, although mapping a real application to the functionality of an actual CF system can be challenging. In any case, here are the broad families of common CF system functionality:

**1) Recommend items**

Show a list of items to a user, in order of how useful they might be. Often this is described as predicting what the user would rate the item, then ranking the items by this predicted rating. However, some successful recommendation algorithms do not compute predicted rating values at all. For example, Amazon’s recommendation algorithm aggregates items similar to a user’s purchases and ratings without ever computing a predicted rating. Instead of displaying a personalized predicted rating, their user interface displays the average customer rating. As a result, the recommendation list may appear out of order concerning the displayed average rating value. In many applications, picking the top few items well is crucial; producing predicted values is secondary.

**2) Predict for a given item**

Given a particular item, calculate its predicted rating. Note that prediction can be more demanding than the recommendation. To recommend items, a system only needs to be prepared to offer a few alternatives, but not all. Some algorithms take advantage of this to be more scalable by saving memory and Computation time. To provide predictions for a particular item, a system. Must be prepared to say something about any requested item, even rarely rated leveraging stem decide how a particular user would rate a requested item if very few users let alone users similar to the particular user have rated it.

**3) Constrained recommendations: Recommend from a set of items**

Given a particular set or a constraint that gives a set of items, recommend from within that set. For example: “Consider the following scenario. Mary's 8-year-old nephew is visiting for the weekend, and she would like to take him to the movies. She Would like a comedy or family movie rated no "higher" than PG-13.

**2.1.2**

We introduce a novel algorithm selection approach, CF4CF, which follows a CF approach to recommend rankings of the most suitable CF algorithms for a new dataset.

The procedure uses the algorithm performance as rating information to induce a metamodel and uses subsampling landmarks converted into ratings to predict the most suitable CF algorithms for a new dataset. According to the experimental results, several conclusions can be drawn:

(1) CF4CF is better at predicting rankings of CF algorithms than standard MTL,

(2) the CF algorithms recommended by CF4CF have a higher impact on the base level performance than those recommended by standard MTL

(3) subsampling landmarks can provide a good initial rating.

Considering that CF4CF uses only performance data and subsampling landmarks to tackle the algorithm selection problem, we have shown that this research direction is a suitable path for CF algorithm selection. Future work directions include: to investigate alternatives to leverage data for training and testing, further extend the experimental setup to other recommendation problems and algorithms, studying the tradeoff between accuracy and time required to create meta-features, and to leverage bot meta-features and ratings in a hybrid CF algorithm selection solution

**2.1.3**

Rapid advances of the Internet and the World Wide Web have greatly facilitated the growth of online applications, such as distance learning, digital library, news on demand, e-commerce, etc. As the availability of these applications continues to increase, users face the tremendous work of retrieving interesting information that matches their preferences. Consequently, users are spending more and more time searching their desired targets and the searching has also drastically increased the consumption of system resources. According to workload analysis of the HPLabs Media server, where 79% of video files belong to a long video group (longer than 30 minutes), 77%-79% of media sessions last less than 10 minutes leveraging long. This implies that many users are not interested in watching the selected videos completely. They may spend a lot of time searching for interesting videos. The demand for efficient and effective tools to help users find their desired targets is required.

Recommender systems provide automated methods for users to search for interesting items with respect to users’ preferences. The underlying techniques used in current recommender systems can be classified into collaborative filtering (CF) and content-based filtering (CBF). CF algorithms exploit similarities among users or items based on users’ feedback. CBF systems, on the other hand, recommend items of interest to the active user by exploiting content information of the items already rated. Typically, a profile is formed for a user individually by analyzing information regarding the content of items, such as desired movies, title minutes leveraging, and description.

**2.2 Integrated Summary**

Collaborative filtering (CF) is a technique used by recommender systems. Collaborative filtering has two senses, a narrow one and a more general one. In the newer, narrower sense, collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if person A has the same opinion as person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person. For example, a collaborative filtering recommendation system for preferences in television programming could make predictions about which television show a user should like given a partial list of that user's tastes (likes or dislikes). Note that these predictions are specific to the user, but use information gleaned from many users. This differs from the simpler approach of giving an average (non-specific) score for each item of interest, for example, based on its number of votes. In a more general sense, collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc. Applications of collaborative filtering typically involve very large data sets. Collaborative filtering methods have been applied to many different kinds of data including sensing and monitoring data, such as in mineral exploration, environmental sensing over large areas or multiple sensors; financial data, such as financial service institutions that integrate many financial sources; or in electronic commerce and web applications where the focus is on user data.