International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

Product Recommendation System Using Machine Learning Based Collaborative Filtering Technique

Mrunal Gandhi[1], Tushar Nikam[2], Akshay Shahane[3], Mayur Shete[4], Dr. Kamini C. Nalavade[5]

[1]Undergraduate Student, Computer Engineering, Sandip Institute of Engineering and Management, Nashik – 422213, India
[1]mkgandhi1195[at]gmail.com
[2] champ96k[at]gmail.com
[3]aakshayshahane19[at]gmail.com
[4]mayurshete3320[at]gmail.com

[5]HOD and Professor, Computer Engineering, Sandip Institute of Engineering and Management, Nashik – 422213, India
kamini.nalavade[at]siem.org.in

Abstract: ***In today's modern epoch of information technology, the idea of efficiently finding one's favourite product in a large dataset of application database, becomes an essential issue to address for the online content providers in order to attract the masses as opposed to their competitors. Recommender systems or recommendation systems, as they are popularly known, are information filtering systems which are usually integrated with several consumer and commercial applications. Such systems act as a bridge between various content facilitators such as social media websites, e-commerce portals, streaming platforms, etc. and the users of these applications, by suggesting them items from the application database which conform to the user preferences and past activities. Such personalized systems play a vital role, especially when the user is unclear of the item to the searched for. These systems are infiltrating every aspect of our lives, in the form of 'Because you watched' header on Netflix, 'People you may know' section on Facebook, 'Customers who bought this also bought' partition on Amazon.***

Keywords: Recommender/recommendation systems, Content based technique, Collaborative technique, Demographic technique, User's preferences, Rating

1. Introduction

Recommender engines are the supporting architectures to the existing commercial software that take in the user footprints as input, analyze them and generate appropriate probabilistic future footprints that the user may be interested in, based on the user's preferences.

One of the distinguishing aspects of personalized recommendation system is that it assists millions of content-consumers limit the quantity of potential items to fit their unique tastes. They help users filter products or services such as books, movies, restaurants, etc. from the humongous alternatives available on the web or in other electronic information sources.

With the engine being fed with a large dataset of items and descriptions of the users' priorities, the internal algorithmic computations then present to the user a relatively narrower set of the items that are well suited to their tailored description. It is this approach of user-oriented comfort and personalization that helps the customers interact better with the system that caters to their needs, while increasing revenues for the virtual businesses that host such systems.

Owing to the fact that recommender engines are data intensive and get better with the increase in content being fed into them, they demand faster complex computing power to perform data analysis on millions of user and product records, and that too, within the least fraction of seconds.

An ideal recommendation system consists of three core components:
1) User resource – analysis of user interests,
2) Item resource – analysis of item features, and
3) The recommendation algorithm.

The user interests are compared with the item features to predict which items to recommend using the recommendation algorithm. The performance of this algorithm is what affects the performance of the whole system.

2. Problem Statement

Multiple existing recommendation engines rely on the content based approach for suggesting data items to their users. However, such a methodology is quite restrictive as the scope of recommendations is limited to a single user's past history and ratings.

Hence, we intend to build a recommender system which is based on a machine learning led collaborative approach. Such systems should generate better explicit outcomes [4], when contrasted with those of content based systems. While content based engines do not prescribe products out of the box and limit the user's choice to explore more, collaborative approach computes the similarity connection amongst several users depending upon their item ratings. Such systems recommend items to other similar users having similar tastes, thus improving the likelihood to explore more out of the organization's product database.

3. Objectives

Usually, the novice recommendation systems consider one of the following techniques for generating suggestions:
a) The preference of user (i.e. content based filtering), or

b) The preference of similar users (i.e. collaborative filtering).

The aim of our project is to build a stable and accurate recommender system using one of many collaborative filtering techniques which should:
a) Suggest accurately similar products by analysing similar minded people's behaviour,
b) Make real-time quality assessment of products by considering other people's experience.

4. Literature Survey

4.1 Movie Recommendation System using Machine Learning [4]

In this paper, the authors stress out on the limitations of content based filtering techniques. Content based systems are constrained to people and do not suggest things out-of-the-box, thus limiting the choice to explore more. Hence, they introduce a model wherein they attempt to combine both content based and collaborative approach.

4.2 Design and Implementation of Movie Recommendation System based on KNN Collaborative Filtering Algorithm [2]

This research paper focuses on the importance of personalized recommender systems. Such systems emerge out as important when the users are unclear about their target content. The authors design and implement system prototype through researching of collaborative filtering algorithms such as KNN. Detailed principle and database architecture model are also included.

4.3 Online Book Recommendation System using Collaborative Filtering (with Jaccard Similarity) [5]

This paper presents the challenges faced by Amazon, Goodreads, etc. to filter, set a priority and give accurate book recommendations. Due to issues of scalability, sparsity and cold start, the authors propose a system combining Collaborative Filtering with Jaccard similarity to give more accurate recommendations.

4.4 Moviemender – A Movie Recommender System [1]

The authors propose to design and implement a hybrid system combining content based and collaborative filtering based recommendation engine. With the objective to provide accurate and efficient movie recommendations, the system has limitations of complexity and higher consumption of resources, making it difficult to implement on a smaller scale of personal computers.

5. Recommendation Techniques

Normally, the recommender systems have the following building blocks [3]:
a) Background data: the existing data that systems already have before recommendation process begins due to past computations,
b) Input data: the data that user feeds in to the system in order to generate recommendations,
c) A blend of recommendation algorithms that combine background data and input data to generate suggestions.

Basically, the techniques by which recommendations are generated, are categorized into five types [3]:
a) Content based filtering,
b) Collaborative filtering,
c) Demographic filtering,
d) Utility based filtering,
e) Knowledge based filtering.

5.1 Content based Filtering Technique [1]

In content based filtering, products are recommended on the basis of comparisons between item profile and user profile. An item profile is a set of pre-assigned keywords for an item, whereas a user profile is a set of pre-assigned keywords collected by algorithm from items found interesting by the user.

A day to day instance of content based filtering technique is as follows: Suppose that a person wishes to buy a pastry from a local baker's shop. Unfortunately, the shopkeeper informs him about the unavailability of the desired pastry and in turn, recommends the person to go for another pastry which has similar characteristics in terms of taste, flavor, price and ingredients. This aptly resembles the content based approach.

Some of the famous online platforms that integrate such an approach are IMDB and Pandora. For this filtering, the technique of Locality-Sensitive Hashing is employed. Locality-Sensitive Hashing (LSH) performs probabilistic reduction of product dimensions or attributes for products having high-dimensional data. A hashing function is used to map the products into similar buckets. The objective is to maximize the probability of collision of similar products. The similarity amongst products is usually calculated using Jaccard similarity technique.

Advantages of content based filtering are:
a) It can recommend unrated items as well,
b) We can easily predict the recommendations by knowing the item profile of an item,
c) It needs only the rating of concerned user to generate suggestions, and not that of other similar users.

Limitations of content-based filtering are:
a) It is inefficient for a recently logged-in user who is yet to rate any product,
b) It does not recommend of unfamiliar items,
c) Limited content is analysed to generate suggestions,
d) It does not recommend items out of the box and hence, limits exploration of the user.

5.2 Collaborative Filtering Technique [1]

Collaborative filtering based systems work on the principle of recommending products by matching the similarity amongst like-minded users and their preferences for certain data items.

A scenario wherein a person asks his friends to recommend him some movies, provided that those friends also have similar likings for a particular genre of movies as that to the person, is a classic example of collaborative filtering technique. Moreover, a recently popularized algorithm is item-to-item collaborative filtering approach, as used by Amazon, an e-commerce giant.

Social networking sites earlier, used pure form of collaborative filtering to recommend new virtual friends and groups by assessing the similarity of connections amongst users and their existing friends.

Advantages of collaborative filtering are:

a) It relies on the similarity of relations between users and hence, is content-independent,
b) Such systems can suggest unfamiliar items by observing similar-minded people's behaviour,
c) They can make real-time quality assessment of products by considering other people's experience.

Limitations of collaborative filtering are [5]:

a) Early rater problem (cold start problem) – Collaborative filtering systems cannot provide recommendations for new added products in the repository, since there are no user ratings yet to calculate a prediction,
b) Grey sheep problem – It is difficult to recommend items to users who do not consistently like or dislike a genre,
c) Sparsity problem – In most cases, most number of active users may only rate a small subset of the overall database. Thus, even the most popular items may have very few ratings and may become ineligible for recommendations,
d) Scalability – With millions of users and products, a large amount of computational power is required to calculate recommendations for large-scale systems.

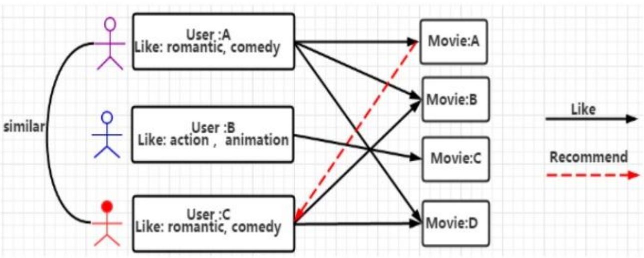

Figure 1: Example of collaborative filtering algorithm [2]

5.3 Demographic Technique [3]

Demographic recommender systems are the engines that classify the users based on personal characteristics and generate recommendations based on the pre-determined demographic classes. An early example of such system was Grundy of 1979 that suggested e-books to users depending on personal data collected via an interactive interface. The personal responses, gathered by a survey, were compared with an existing database of already assembled user classes. The advantage of such an approach is that it does not need historical data of user's past ratings unlike the collaborative and content-based techniques.

6. Proposed System Methodologies

This section elaborates on the technical methodologies to implement the proposed system, along with a due consideration given to the sequence of steps. The proposed system is also explained with respect to its operations, functions, and events.

6.1 KNN Algorithm [2]

KNN algorithm stands for K-Nearest Neighbor algorithm. The principle of this algorithm is that: if a majority of the k-most similar neighbors (points) of a sample point belongs to a certain category in the problem space, then the sample point is also considered to belong to this category.

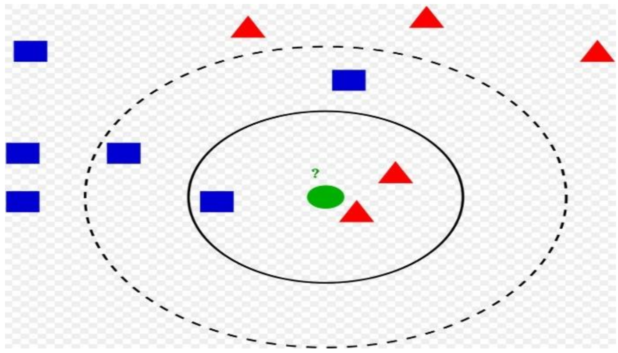

Figure 2: Illustration of KNN algorithm

Suppose that the green circle is the sample point, for which the most similar nearest neighbors are to be found. If k = 3 (solid line circle), then the green sample point is assigned to the majority red triangles' category as there are 2 red triangles and only 1 blue square. If k = 5 (dotted line circle), then the green sample point is assigned to the majority blue squares' category as there are 3 blue squares and only 2 red triangles.

6.2 Collaborative Filtering based KNN Algorithm [2]

KNN collaborative filtering algorithm is a collaborative filtering based KNN algorithm. It uses KNN algorithm to search, analyze and choose the most similar nearest neighbors. The sequence of steps to be implemented under this algorithm are:

a) User similarity calculation,
b) KNN nearest neighbour selection, and
c) Predict score evaluation.

6.2.1 User Similarity Calculation

Consider the following user-product dimensionality matrix. Here, we consider the user set, U = {u1, u2, u3, u4} and the movie set, M = {m1, m2, m3, m4}.

U\M	m1	m2	m3	m4	m5
u1	1	3	3	4	2
u2	3	1	4		
u3	2	4		1	5
u4	2		2		

Figure 3: User-product matrix [2]

Paper ID: SR201127214109 DOI: 10.21275/SR201127214109

The values filled in the matrix represent the corresponding ratings given by each user to each movie. For calculating the similarity between users, we need to have their respective rating vectors. Say, for example, we were to calculate the similarity between users u1 and u3. Then their corresponding vectors are u1 = {1, 3, 4, 2} and u3 = {2, 4, 1, 5}.

The similarity between two users x and y is often calculated using:

(a) Cosine Similarity
To calculate the similarity, cosine of angle between two vectors is used.

$$sim(x,y) = \cos(\vec{X},\vec{Y}) = \frac{\vec{X} * \vec{Y}}{|\vec{X}| * |\vec{Y}|} = \frac{\sum_{s\in s(x,y)} r(x,s) * r(y,s)}{\sqrt{\sum_{s\in s(x,y)} \left[r(x,s)^2\right]}\sqrt{\sum_{s\in s(x,y)} \left[r(y,s)^2\right]}} \quad (1)$$

Here,
r(x,s) = rating by user x to product s,
r(y,s) = rating by user y to product s,
s(x,y) = set of movies that both users rated.

(b) Pearson Correlation Similarity
To calculate the similarity, linear relation between two vectors is used.

$$sim(x,y) = \frac{\sum_{s\in s(x,y)} \left[r(x,s) - \overline{rx}\right]\left[r(y,s) - \overline{ry}\right]}{\sqrt{\sum_{s\in s(x,y)} \left[r(x,s) - \overline{rx}\right]^2}\sqrt{\sum_{s\in s(x,y)} \left[r(y,s) - \overline{ry}\right]^2}} \quad (2)$$

Here,
$\overline{rx}$ = average rating by x,
$\overline{ry}$ = average rating by y.

6.2.2 KNN Nearest Neighbor Selection

Consider the following illustration for the selection of k-nearest neighbors, where k = 7 and the red point is the sample data item for which the most similar 7 neighbors are to be calculated.

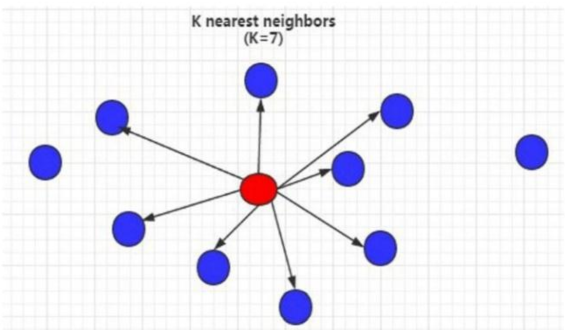

Figure 4: Illustration of KNN for k = 7 [2]

6.2.3 Predict Score Evaluation

The average rating score calculation for a data item 'i' is given by,

$$r(u,i) = \overline{ru} + k\sum_{u'\in U} sim(u,u') * \left[r(u',i) - \overline{ru'}\right] \quad (3)$$

$$k = 1/\sum |sim(u,u')| \quad (4)$$

Here,
r(u,i)= predicted rating of item 'i' to user 'u'.

6.3 Mathematical Model

Consider the proposed system to be a set of tuplets such that:
S = {I, O, F, Success, Failure}, where
S = Proposed system,
I = Set of inputs to the system,
O = Set of outputs from the system,
F = Set of system functions,
Success = Success case,
Failure = Failure case.

Here,
I = {I1, I2, I3}, where
I1 = User credentials, I2 = User ratings, I3 = Addition of products to existing database by the admin,
O = {O1, O2}, where
O1 = Generating recommendations, O2 = Displaying recommendations,
F = {F1, F2, F3}, where
F1 = User similarity calculation, F2 = KNN nearest neighbor selection, F3 = Predict score evaluation,
Success = Generating feasible and optimum product recommendations based on collaborative filtering technique,
Failure = Not generating recommendations based on collaborative filtering technique.

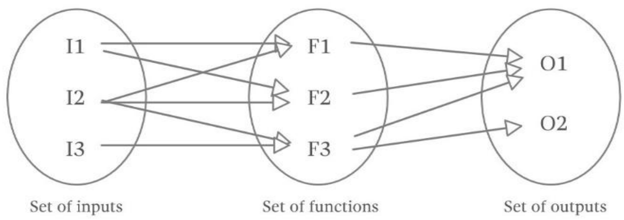

Figure 5: Proposed mathematical model

6.4 Proposed System Architecture

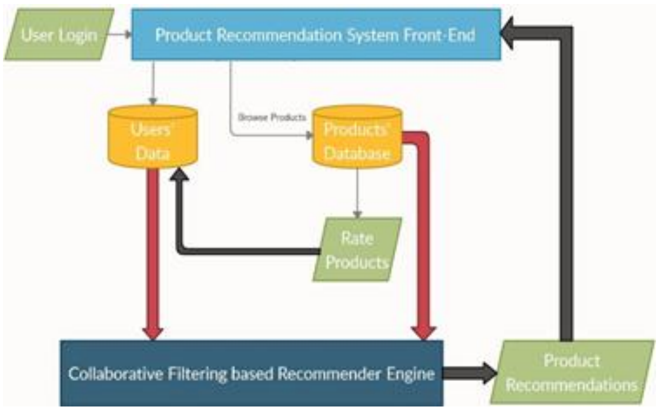

Figure 6: Proposed system architecture

7. Acknowledgement

We express our heartfelt gratitude to all the well-wishers and benefactors involved in assisting us to propose the specified model and its related architecture. Furthermore, we extend our acknowledgements to our research paper guide, Dr. Kamini C. Nalavade, under the mentorship of whom our potential endeavors witnessed their ultimate realities. It is to the authors of our literature survey and their respective notable publishing institutions that we are immensely grateful to.

8. Conclusion

Since the mentioned recommendation system is proposed to be based on machine learning led collaborative filtering technique, its outcomes are expected to be explicitly different from systems that use content based approach. Our system is supposed to calculate the similarities amongst numerous product retail application users, keeping in view their similar tastes, and then by analyzing their ratings for the data products, recommending items to other such similar users, thus allowing them to explore more of the commercial repository.

References

[1] RupaliHande, AjinkyaGutti, Kevin Shah, Jeet Gandhi, VrushalKamtikar,“Moviemender – A Movie Recommender System,” International Journal of Engineering Sciences & Research Technology, 2018.
[2] Bei-Bei Cui,“Design and Implementation of Movie Recommendation System based on KNN Collaborative Filtering Algorithm,” ITM Web of Conferences, ITA 2017.
[3] AbhilashaSase, KritikaVarun, SanyuktaRathod, Prof. DeepaliPatil,“A Proposed Book Recommender System,” International Journal of Advanced Research in Computer and Communication Engineering, 2015.
[4] F. Furtado, A. Singh,“Movie Recommendation System using Machine Learning,” International Journal of Research in Industrial Engineering, 2020.
[5] AviRana, K. Deeba,“Online Book Recommendation System using Collaborative Filtering (with Jaccard Similarity),” International Conference on Physics and Photonics Processes in Nano Sciences, 2019.

Author Profile

Mrunal Gandhi (mkgandhi1195@gmail.com) is a final year Computer Engineering undergraduate student from Sandip Institute of Engineering and Management, a NAAC B++ accredited institute affiliated to the SavitribaiPhule Pune University and situated in Nashik. He has been extremely consistent in his academic performance with a 9.29 cumulative grade point average until the termination of 6 semesters of his BE studies.