Crowdfunding Platform Recommendation Algorithm Based on Collaborative Filtering

DOI : 10.36909/jer.ICCSCT.19487

Yaming Li[1], Jiahao Liu[1], Yuying Jin[2], Xinxin Fan[3], Bixi Wang[4]*

[1]Shaanxi University of Science and Technology, Xi'an, 710016, China.
[2]Xi'an International Studies University, Xi'an, 710119, China.
[3]University of Electronic Science and Technology of China, Chengdu, 610054, China.
[4]Xi'an University, Xi'an, 710065, China.

* Corresponding Author: wangwbx@xawl.edu.cn(BW)

ABSTRACT

Crowdfunding platforms are a novel method of internet fundraising. The production of data from the crowdfunding platform has increased, but the benefits of this data have not. This has led to a state of "information overload." User-specific recommendation systems driven by data mining can play an effective role in solving this problem. This study introduces a collaborative filtering system that is driven by its end-users by fusing user input with the closest neighbor technique from machine learning. The results of this test demonstrate that the algorithm can provide useful project suggestions to those who utilize the crowdfunding site.
Keywords: Personalized recommendation; Crowdfunding platform; Machine learning; Collaborative filtering.

INTRODUCTION

From the concept of the recommendation system to now, its exploration and implementation have made great progress (Das et al., 2017, Kumar et al., 2018, Cui, Y., 2021 & Da'u et al., 2021). ACM has held many academic conferences with recommended technology as the research theme, and many well-known journals have also taken it as a research hotspot (Hamid et al., 2021, Hui et al., 2022, Ihnaini et al., 2021, Mohammadi et al., 2022 & Djenouri et al., 2022). The Netflix competition from 2006 to 2009 (Koren. Y., 2009) made the research of recommendation technology more popular. Due to the continuous emergence of various network platforms, recommendation systems also appear on many types of websites. For a relatively complete recommendation system, its three main components include the behavior record component, user modeling component, and recommendation algorithm component (Roshni et al., 2015), which are mainly used to collect user information and construct user preference models (Isinkaye et al., 2015).
(1) The main goal of the user's online behavior history module is to gather and record the user's prior activities.
(2) The main function of the user modeling module is to take the information of the behavior recording module as input and calculate the similarity between users through relevant formulas or algorithms.
(3) The recommendation algorithm module will weigh and calculate the user's possible

interest in an item according to the similarity between users and the preference of these similar users for the item. Generate a push service according to the ranking of his interest.
Every specialized recommendation system must have a recommendation algorithm module (Ko et al., 2022). The reliability of the guidance will improve instantly as a result of this module. Collective filtering recommendations and content-based recommendations are the most discussed and debated types of recommendation algorithms.

The development of content-based recommendation systems is propelled by studies in information-filtering technology. The system analyzes the user's prior preference item using text analysis and feature extraction to obtain characteristics that may identify the attribute of the item. This is carried out by the data stored in the user's history preference item. Find items with comparable text qualities among all the goods to be offered, and then propose the item with the greatest similarity score to the user. In this technique, the system is not required to know the user's past performance on certain goods. In addition, the user is not required to complete the survey form on item preferences, etc. Using the item's historical data and information on its content, the recommendation may be completed immediately.

The most often cited implementation of a content-based recommendation system is the fab system (Balabanovic et al., 1997). The system's primary goal is to suggest sites that you may like browsing. The system analyses each page to discover and extract attributes, and it uses 128 distinct keywords as web page characteristics. The subsequent stage is to extract the web page characteristics from the user's access data, identify the online sites that are most comparable to the user's desired qualities, and then recommend those web pages to the user.
The greatest difficulty offered by content-based recommendation technology is feature extraction and information filtering, both of which have a direct bearing on the accuracy of suggestions. In recent years, owing to the efforts of a large number of scholars in this subject, its theory has advanced significantly. The vector space model based on TF-IDF weight is the most often used.

When the technology of collaborative filtering originally arose, it was important for the recommendation system to enable users to actively collect one another's preference information before proceeding with the prediction and pushing processes. In response to the increasing social need, researchers are devoting more time and effort to this aspect. Following a period of study, there is now an independent recommendation service that is based on collaborative filtering. A typical use of this service is the group lens system (Resnick et al., 1994). The group lens was created by the Minnesota State University team working on the group lens project. Before anything else, the system constructs a data model, for which it solicits information about the users' preferences. In the next step, the system examines the data model and determines the level of similarity between the users. To conclude the recommendation service, the tried-and-true approach of collaborative filtering is implemented.

Compared with the content-based recommendation, the collaborative filtering algorithm does not need to perform text analysis and feature mining on user preference items, but completes recommendations by acquiring potential user preferences, which has the advantage of customization. This is also the main reason why collaborative filtering technology has attracted so many researchers' attention and is widely used in e-commerce platform recommendation and other fields.

We create a personalized recommendation system for crowdfunding platforms using collaborative filtering in this study. The closest neighbor algorithm may be thought of as the bare bones of this machine learning technique. Crowdfunding platform user score data is gathered to test and evaluate the algorithm's viability in light of the platform's specific data features.

DATASET AND METHODS

2.1 Dataset

The experimental data are selected from the following perspectives:
(1) The scenario studied in this paper is a crowdfunding platform, and the input data required for algorithm verification should also be network crowdfunding data. These data should be generated by the user's behavior on the crowdfunding platform.
(2) The algorithm used in this research was built using the collaborative filtering technique with human users. To create this algorithm, it is necessary to gather information about users, items, and users' ratings of those items.
It is up to us to compile the data required for this investigation. After looking into a large number of domestic and international crowdfunding sites, it was discovered that roll call time had evaluations from actual users. Roll Call Time's rise to prominence may be attributed to the high-quality crowdfunding solutions it offers. Indicative of their generalizability is the large sample size of both consumers and projects. To fine-tune and verify the algorithm, we will collect user and project data from the roll call time platform as this article progresses.
You may learn more about the 1095 crowdsourcing projects that were completed by visiting the "roll call time" webpage. The platform categorizes these endeavors into the following nine groups: film and television; advertising display; digital communication; home life; smart clothing; video and audio entertainment; travel positioning; the arts and culture; and the culinary arts. The 277 items that were assessed received a total of 4645 user ratings, with 3445 persons taking part in the scoring process. The users will rate the project on numerous dimensions, such as originality, design, and usability, and the ultimate value will be determined by arithmetic mean. Considering that the great majority of users have contributed just a few remarks, the user rating matrix is quite sparse.
In conclusion, it was revealed that users of the roll call time website supported a total of 1084 projects, with 172841 individuals supporting the projects, 88153 users engaging in the support, and 89427 accessible platforms in total.
The collaborative filtering algorithm is comprised of two basic building pieces, which are user-based collaborative filtering suggestions and item-based collaborative filtering recommendations. The user-based collaborative filtering algorithm is the essential component of the recommendation system that is being investigated in this project. These results are based on the findings of previous research that used a crowdsourcing platform as its primary data source.
Users are considered to be neighbors with one another if they have a high degree of similarity and their history score records are located within proximity to one another. The fundamental idea behind the algorithm (Ajaegbu et al., 2021) is that it will decide that two individuals are neighbors to each other if they have a lot of qualities in common. Because of this, the

algorithm can make an educated guess as to the rating that the second user will give the item based on the rating that was given by the first user. This is particularly helpful for items that are seldom reviewed. The last step is to arrange the ratings in such a way that the user is supplied with suggestions for the best n things. It is possible to determine the projected score of a user for the whole system by giving the scores of a certain number of similar users a specified degree of weight.

2.2 The collaborative filtering algorithm

Techniques such as collaborative filtering (CF) provide consumers recommendations that are specifically catered to their preferences and are derived from the ratings and activities of other users (such as purchases). Figure 1 depicts a typical example of the collaborative filtering approach being used throughout the proposal process.

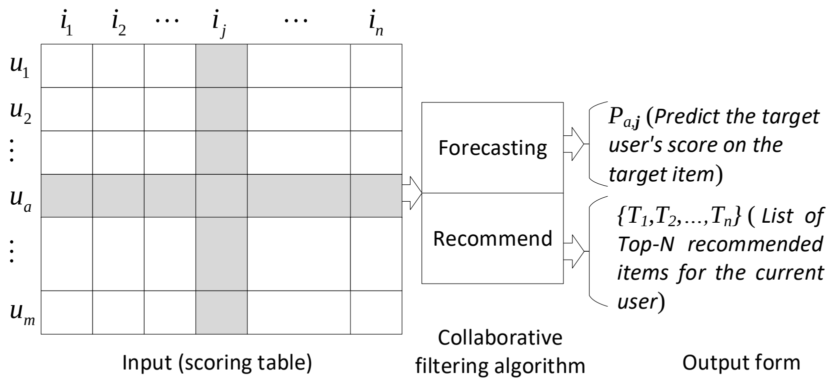


Figure 1. Flow chart of collaborative filtering algorithm recommendation.

The following is the specific flow of the user-based collaborative filtering algorithm (Zhang et al., 2020):

2.2.1 Data representation

As stated in Table 1, an example of input is information supplied by the user to the algorithm in the form of a score for the item.

Table 1. User item scoring data.

Item ID / User ID	Item_1	Item_2	⋯	Item_n
User_1	$r_{1,1}$	$r_{1,2}$	⋯	$r_{1,n}$
User_2	$r_{2,1}$	$r_{2,2}$	⋯	$r_{2,n}$
⋯	⋯	⋯	⋯	⋯
User_m	$r_{m,1}$	$r_{m,2}$	⋯	$r_{m,n}$

A matrix representation of the data is required for the user similarity computation to be finalized. In this model, the user occupies one column and the object occupies another. A user's evaluation of an item is represented by a value in the matrix. The number of elements in the matrix is. The formula (1) for scilicet shows that there are both users and goods.

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{bmatrix} \quad (1)$$

2.2.2 Neighbor query

In this stage (Herlocker et al., 1999), finding the users closest to the user is the primary focus. Similar interests and preferences are assumed when two users have score records for the same item and their values are quite close to one another. The concept of "neighbor selection" relies on this premise. The closest neighbor query must first compare every user to determine the degree of similarity between them. Since this similarity would have an immediate effect on the efficacy of the recommendation algorithm, the technique of calculating similarity is also an important research subject that needs to be explored. Here is a rundown of some common approaches the algorithm should use while looking for user-submitted similarity solutions:

(1) Cosine similarity. The user's scores on all items are used to represent the user's preferences. For non-scored items, the score is generally set to 0 and the vector is filled. Each user can be represented by a vector of length n (i.e., the row vector in R). For users u and v, their preference vectors are $\vec{u}$ and $\vec{v}$ respectively. The similarity between users $\vec{u}$ and $\vec{v}$ can be represented by calculating the cosine values of u and v, as shown in formula (2).

$$sim(u,v) = cos(\vec{u},\vec{v}) = \frac{\vec{u}\vec{v}}{\|\vec{u}\|\times\|\vec{v}\|} \quad (2)$$

(2) The bond between them (Pearson correlation coefficient). User ratings, and finds the items in the user scoring matrix where the user scored anything other than zero in the -TH row. represents the user-rated items and clears the -TH row of the user score matrix of all values other than zero. represents the sum of all scores given by both the user and the user. If you want to know how closely you and another user are alike, you may use the formula (3) below.

$$sim(u,v) = \frac{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_{uv}}(r_{v,i}-\bar{r}_v)^2}} \quad (3)$$

(3) Similarity measured via a cosine transform (Spearman correlation coefficient). User similarity, as measured by the Spearman correlation coefficient, takes into account the average value of users' scores for items since everyone has their criteria for grading the same thing. Following the formula (4), this technique determines the user's deviation vector about his mean score.

$$\mathrm{sim}(u,v) = \frac{\sum_{i\in I_{uv}}(r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I_u}(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_v}(r_{v,i}-\bar{r}_v)^2}} \quad (4)$$

In the user-based collaborative filtering method, a comparison of the three aforementioned schemes found that Pearson's correlation coefficient was the most effective. The formula (5) for computing the user similarity matrix is used throughout this investigation.

$$SIM = \begin{bmatrix} sim(1,1) & sim(1,2) & \cdots & sim(1,m) \\ sim(2,1) & sim(2,2) & \cdots & sim(2,m) \\ \vdots & \vdots & \ddots & \vdots \\ sim(m,1) & sim(m,2) & \cdots & sim(m,m) \end{bmatrix} \quad (5)$$

2.2.3 Generate recommendations

Both the scoring matrix R and the user similarity matrix SIM are used to make a prediction about the user's items and the scores they should be suggested with. Arrange the prediction scores from highest to lowest, and then choose the products that should be recommended to the user in ascending order. The process of producing prediction scores will be broken out in the following paragraphs.

The K neighbor users of a user are the ones that are chosen from the similarity matrix to construct the user's neighbor set KNB_u.. The anticipated score of the user KNB_u. for the item may be determined using the equation (6) below, which takes into account the degree of similarity between the item i to be suggested and the user in KNB_u question, as well as the user's u actual score $\hat{r}_{u,i}$ value for the item i in question.

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{n \in KNB_u} sim(u,n) \times (r_{n,i} - \bar{r}_n)}{\sum_{n \in KNB_u} (|sim(u,n)|)} \tag{6}$$

Where indicates the intended user, r represents the average score of all items, represents a user in the nearest neighbor set of the target user, represents the score similarity between the target user and the nearest neighbor user, represents the score value of the target user on the item, and represents the average score of on all items.

The flow of customized recommendation algorithm of crowdfunding platform is displayed in figure 2.

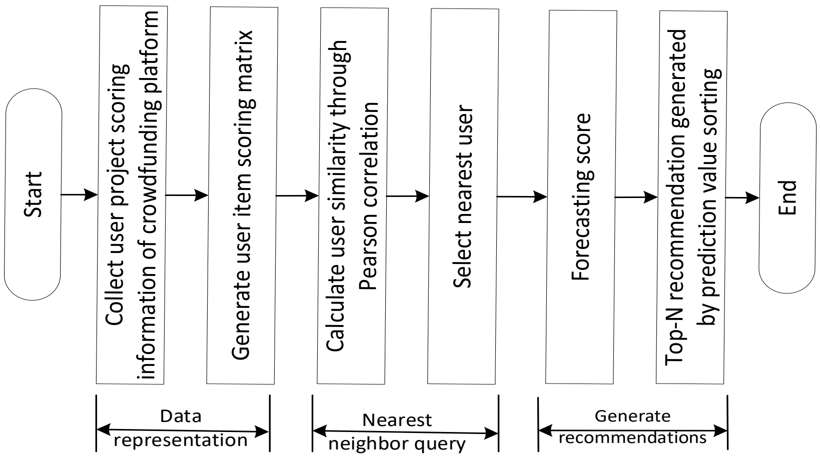


Figure 2. Flow chart of crowdfunding platform recommendation algorithm based on collaborative filtering.

EXPERIMENTS AND RESULTS

3.1 Evaluating indicator

The performance of the system is represented by the evaluation index. When it comes to the recommendation system, consumers are most concerned with whether or not the recommended products match the desired items and whether or not the accuracy of the suggestions is sufficient. The statistical accuracy metric and the decision support metric are the two most important evaluation metrics for the recommendation system (Sarwar et al., 2001).

(1) A method of measuring that places a premium on statistical reliability. Most of the evaluation for this method is dependent on statistical expertise, which is often straightforward to compute. The technique's merits and flaws may be assessed by comparing the mistakes that occurred before and after the forecast. The two most common statistical techniques are the mean absolute error (MAE) and the root means square error (RMS) (RMSE). The average

absolute error may be calculated using the following formula (7).

$$MAE = \frac{1}{|\tau|} \sum_{(u,i)\in\tau} \left|\hat{r}_{u,i} - r_{u,i}\right| \quad (7)$$

The user's projected score and actual score on each item in the test set are represented by $\hat{r}_{u,i}$ and $r_{u,i}$ i, respectively, where τ is the test sis absoluteis$|\tau|$ is the total number of items in the test set, and is the size of the test set.

Following is the formula (8) for determining root mean square error RMSE .

$$RMSE = \sqrt{\frac{1}{|\tau|} \sum_{(u,i)\in\tau} (\hat{r}_{u,i} - r_{u,i})^2} \quad (8)$$

The final performance evaluation of the system will be based on the average absolute error, which will be used in this research. The lower the MAE is, the more exact the prediction score and persuasive the recommendation.

(2) A method for evaluating the accuracy of judgments. This method analyzes the system from a unique perspective, and the results show two separate aspects: those that are accurate and those that are incorrect. The evaluation method is basic. If the user's favorite item is included in the result of the proposal, the recommendation was successful; otherwise, it was not. For a recommendation system with a maximum score of 5, the threshold is set at 3.5 points, for example. If the user-suggested inter-item score is more than 3.5 points, this shows that the advice was successful, and the algorithm does not care how much higher the score is.

There may be four situations in the project to be predicted, and the number of projects to be predicted in these four situations is indicated by N_{tp}, N_{fn}, N_{fp} and N_{tn} respectively, the letter *t* means "Ture", the letter *f* means "False", the letter *p* means "Positive", and the letter *n* means "Negative", as shown in Table 2.

Table 2. Classification of items to be predicted.

	System recommended	System didn't recommended
Users like	N_{tp}	N_{fn}
Users doesn't like	N_{fp}	N_{tn}

The primary data was partitioned into a training set and a test set. Predictions are made based on the training set, and the performance of the system is assessed by comparing the projected score to the actual scores of users in the test set. There are two primary metrics used for analysis: accuracy and recall.

The system creates a list of recommendations via analysis and computation, and this is used as the basis for the calculation of the recommendation accuracy rate. By dividing the predicted number of right recommendations by the total number of suggestions, we may get a sense of the recommendation accuracy rate. The calculation with formula (9) shows that this fraction is equivalent to the percentage of the first item in the first column of Table 2.

$$Precision = \frac{N_{tp}}{N_{tp} + N_{fp}} \quad (9)$$

To get the user's Recall, multiply the percentage of the first item in the first row of Table 2 by the percentage of the entire number of things the user liked from the suggestion list. The calculation formula (10) is shown below.

$$Recall = \frac{N_{tp}}{N_{tp} + N_{fn}} \tag{10}$$

Once you learn how to compute the two, you'll see that they don't rise or fall together, and in fact, they often go in the other way. Designing an index that holistically balances advances in accuracy and recall is necessary for improving both metrics. The F-measure is a common statistic, and its formula (11) is given.

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

F-measure provides a more complete picture of the system's behavior. The greater the number of users, the more accurate and personalized the recommendations will be.

3.2 Experimental and result

For the purposes of this article, the Java programming language will be used to fully implement the algorithm. The experiment employs a variable-control technique to alter the ratio of training data to test data, and then evaluates the feasibility of the algorithm developed in this study by examining its average absolute error, accuracy, recall, and f index.

3.2.1 Experiment 1

Crowdfunding site ratings are split between a "training set" and "test set" to optimize the system. By analyzing the data from the training set, a projected score for the test set may be generated. In order to minimize experimental error, it is important to monitor the size of the training set in relation to the size of the test set on a regular basis. We implement the user-based collaborative filtering algorithm with the default number of neighbors after adjusting the training set to test set ratio and calculating the average absolute error of the testing method. In order to account for external factors like random sampling and hardware conditions, as shown in Figure 3, five separate trials were conducted for each parameter. The mean value was then used.

Figure 3. MAE value of collaborative filtering algorithm when the proportion of training set and test set is different.

In Figure 4, we see a visual representation of the evolution of this ratio as a function of MAE both the training set and the test set proportions, from which we may extract a trend diagram depicting this evolution.

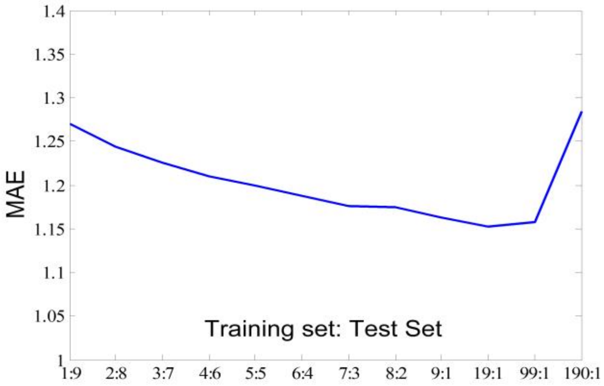


Figure 4. Effect of the ratio of training set and test set on the system.

The performance of the system is very sensitive to the total quantity of data used in the training process, as seen by the curve in the preceding paragraph. The MAE value decreases as the ratio of the training set to the test set is larger, and this ratio might be anywhere from 1:9 to 19:1. Assuming a minimum value of 19:1, the ratio of training samples to test samples is considered optimal. However, it grows dramatically when the proportion of the test set to the training set is more than 99 to 1. This is because there will be a great deal of variation in the assessment indicators if the data from the test set is insufficient, meaning that the results will not be representative and individual forecasts will be wrong.

The future experiments will utilize a training set 19 times larger than the test set used to evaluate the approach in order to confirm this hypothesis. Better accuracy is now possible.

3.2.2 Experiment 2

According to Figure 5, the nearest neighbor count may be adjusted within the range [5,95], with an increment of 5. Take note of the fluctuation in cost. Iteratively increase the number of users' nearest neighbors by 5 each time using a training set comprised of 95% of the available score data (19:1 training to test ratio).

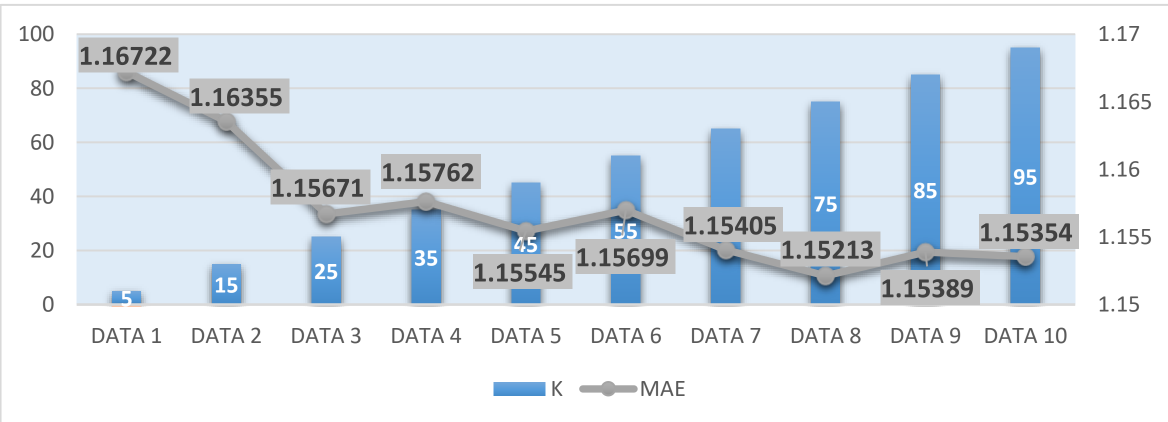


Figure 5. "MAE" value of collaborative filtering algorithm when the number of neighbors is

different.

If the nearest neighbor number K is the abscissa and the MAE value is the ordinate, the MAE value change curve is shown in figure 6.

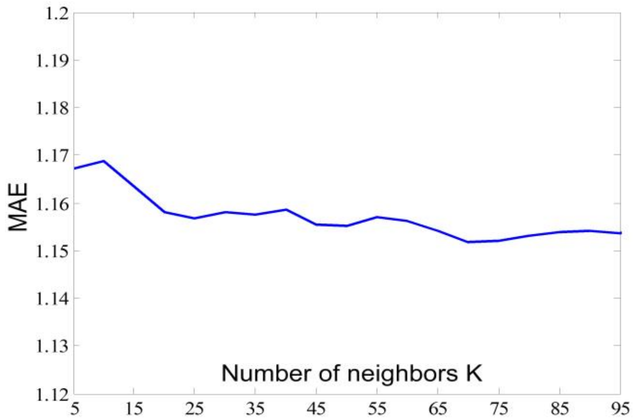
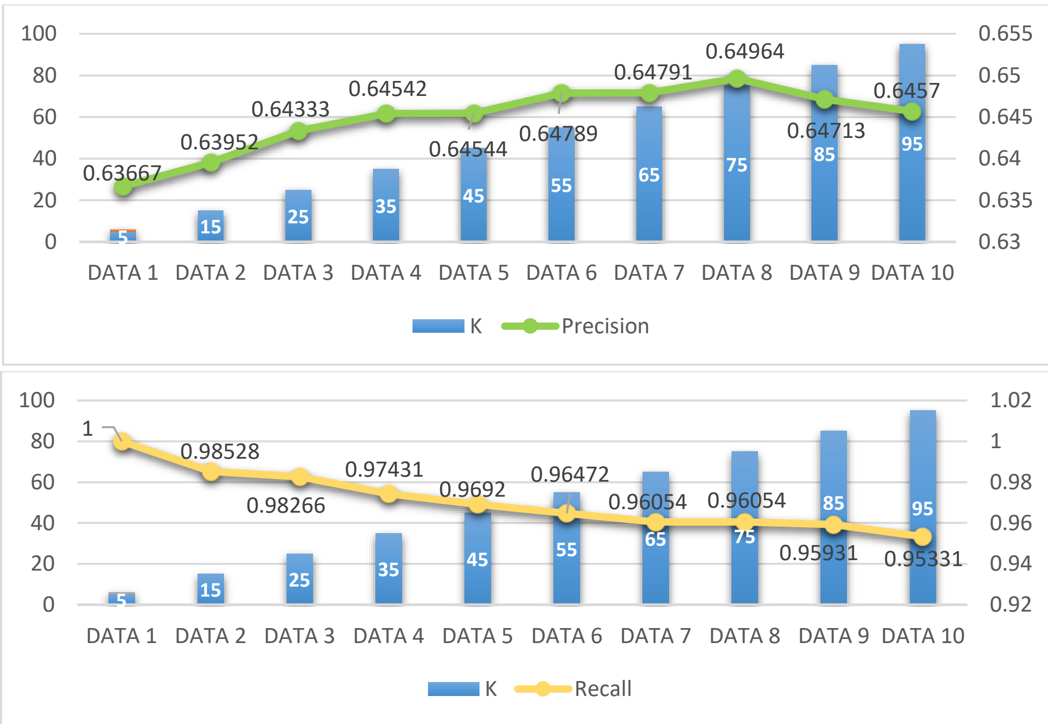


Figure 6. Collaborative filtering's utility varies with the number of neighbors.

For the user to have given the project a score of 3.5 or above, their enthusiasm for it must be shown. Evaluate the accuracy and recall of the method, and then use the combined scores to calculate the f index, as shown in Figure 7.



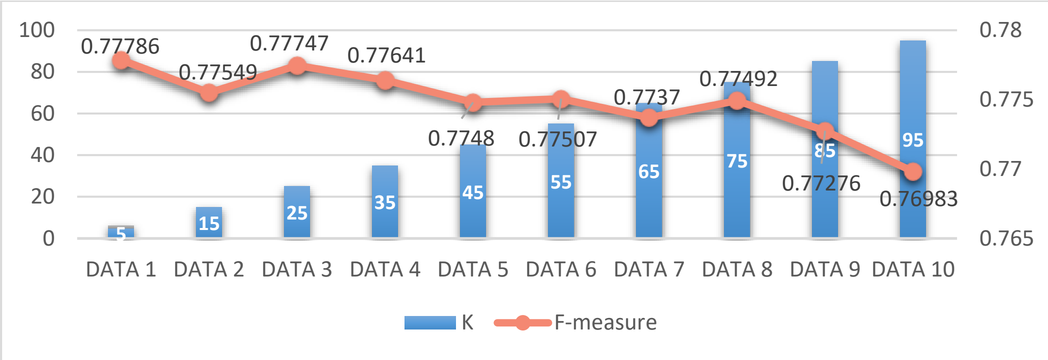


Figure 7. Evaluation indexes of collaborative filtering algorithm when the number of neighbors is different.

The change of F-measure with the number of neighbors K is expressed in the coordinate system as shown in Figure 8.

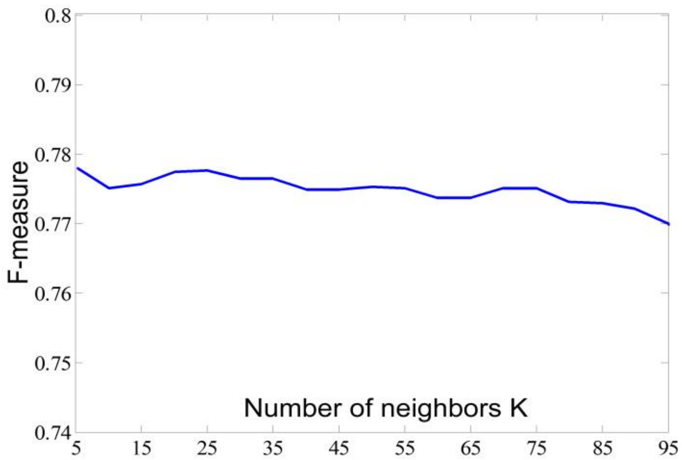


Figure 8. Change of F-measure value of collaborative filtering algorithm with the number of neighbors.

DISCUSSION AND CONCLUSION

The MAE and F-measure tests provide light on the algorithm's inherent characteristics in the following ways:

(1) The algorithm that is based on collaborative filtering and is used for creating suggestions on the crowdfunding site has an average absolute error of 1.15 percentage points.

(2) The forecasts have an accuracy of more than 0.6 despite their low score.

(3) There is a lack of clarity on the manner in which the neighbor count influences the effectiveness of the algorithm: Given these characteristics, we are able to draw the following conclusions

(1) The crowdsourcing platform is compatible with the collaborative filtering approach proposed in this research.

(2) The cold start difficulty and sparseness of the data matrix are major contributors to the method's large amount of prediction inaccuracy.

As part of our project, we developed a custom recommendation algorithm for a crowdfunding website by combining the collaborative filtering algorithm with the closest neighbor approach of machine learning. The algorithm's three most crucial operations are data collection, similarity computation, and recommendation creation. We analyzed the crowdsourcing website's user evaluations to determine the algorithm's effectiveness. It makes sense to utilize web crawlers to collect customer reviews from crowdfunding platforms (the website data of roll call time is used in this paper). To assess the efficacy of the algorithm, choose an experimental evaluation index and an evaluation metric (F-measure, MAE, or something similar). At the start, we look at the ideal test set to training set ratio (in this case, 19:1). The next step in evaluating the approach's viability and the difficulties it already poses is to keep track of the output Mae and F-measure indices while adjusting the number of neighbors. Both a lack of information and a slow start are major issues.

REFERENCES

Das, D., Sahoo, L. & Datta, S., 2017. A survey on recommendation system. International Journal of Computer Applications.160 (7).

Kumar, P. & Thakur, R.S., 2018. Recommendation system techniques and related issues: a survey. International Journal of Information Technology. 10.495–501.

Cui, Y., 2021. Intelligent recommendation system based on mathematical modeling in personalized data mining. Mathematical Problems in Engineering.2021 (2021).

Da'u, A., Salim, N. & Idris, R., 2021. An adaptive deep learning method for item recommendation system. Knowledge-Based Systems.213 (2021): 106681.

Hamid, R.A., Albahri, A.S., Alwan, J.K., Al-qaysi, Z.T., Albahri, O.S., Zaidan, A.A., Alnoor, A., Alamoodi, A.H. & Zaidan, B.B., 2021. How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management. Computer Science Review.39 (2021): 100337.

Hui, B., Zhang, L., Zhou, X., Wen, X. & Nian, Y., 2022. Personalized recommendation system based on knowledge embedding and historical behavior. Applied Intelligence.52: 954–966.

Ihnaini, B., Khan, M.A., Khan, T.A., Abbas, S., Daoud, M.S., Ahmad, M. & Khan, M.A., 2021. A smart healthcare recommendation system for multidisciplinary diabetes patients with data fusion based on deep ensemble learning. Computational Intelligence and Neuroscience.2021 (2021).

Mohammadi, N. & Rasoolzadegan, A., 2022. A two-stage location-sensitive and user preference-aware recommendation system. Expert Systems with Applications.191 (2022): 116188.

Djenouri, Y., Belhadi, A., Srivastava, G. & Lin, J.C.W., 2022. Deep learning based hashtag recommendation system for multimedia data. Information Sciences.609 (2022): 1506-1517.

Koren, Y., 2009. Collaborative filtering with temporal dynamics. Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.2009: 447–456.

Roshni, K.S. & Sachin, N.D., 2015. Comparative study on approaches of recommendation system. International Journal of Computer Applications.118. 2 (2015).

Isinkaye, F.O., Folajimi, Y.O. & Ojokoh, B.A., 2015. Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal.16 (3).

Ko, H., Lee, S., Park, Y., Choi, A., 2022. A survey of recommendation systems: recommendation models, techniques, and application fields. Electronics.11(1):141.

Balabanovic, M. & Shoham, Y., 1997. Fab: content-based, collaborative recommendation. Communications of the ACM.40 (3): 66-72.

Resnick, P., Iacovou, N. & Suchak, M., 1994. GroupLens: an open architecture for collaborative filtering of netnews. Proceedings of the 1994 ACM conference on Computer supported cooperative work. 1994: 175-186.

Ajaegbu, C., 2021. An optimized item-based collaborative filtering algorithm. Journal of Ambient Intelligence and Humanized Computing.12 (12).

Zhang, Z., Peng, T., & Shen, K., 2020. Overview of collaborative filtering recommendation algorithms. IOP Conference Series: Earth and Environmental Science.440 (2020) 022063.

Herlocker, J.L., Konstan, J.A. & Borchers, A., 1999. An algorithmic framework for performing collaborative filtering. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. Berkeley: 230-237.

Sarwar, B., Karypis, G. & Konstan, J., 2001. Item-based collaborative filtering recommendation algorithms. International Conference on World Wide Web. Hong Kong:285-295.