**User-Based Collaborative Filtering in Recommendation Systems**

Anoushka Saha, Anshika Srivastava, Arshiya Srivastava, Deeksha Mishra, Deepali Rajput, Harshita Pahwa, Khushi Meemrot and Yoshita Sood

DSKC Summer Workshop 2020

Miranda House, University of Delhi

**ACKNOWLEDGEMENT**

User-Based Collaborative Filtering in Recommendation Systems

The present project report deals with the topic "User-Based Collaborative Filtering in Recommendation Systems" under the supervision and guidance of the Department of Mathematics.

We express our deep gratitude to our mentors Mr Naveen Gupta and Mr Mohit Kumar Macchal for their guidance and help in completing the project in time. We are thankful to everyone concerned with this project, without whose guidance our project would have lacked the necessary information needed to complete our purpose of study.

Dated: July 2020

**ABSTRACT**

User-Based Collaborative Filtering in Recommendation Systems

The existing cluster of websites like Facebook, Amazon, and Netflix caters carefully calculated recommendations to users based on pre-existing user-items relationships depending on the problem addressed.

         A recommendation system includes algorithms that can predict what a user may like or dislike from a set of given items or it can rely on the likes and dislikes of other similar users. The results obtained from the system can then be used in a diverse collection of applications.

This thesis particularly aims at constructing a recommendation system that predicts the likelihood of a particular user liking a specific movie from a default movie dataset. Multiple users’ ratings’ of movies are extracted from the dataset using Python. If we choose a particular user then the system returns a set of movies that the selected user might like with the help of the predicted ratings generated by the program.

User-Based Collaborative Filtering in Recommendation Systems

TABLE OF CONTENTS

[INTRODUCTION 1](#_Toc47894361)

[OBJECTIVE 3](#_Toc47894362)

[METHODOLOGY 4](#_Toc47894364)

[1. User-Based Collaborative Filtering 4](#_Toc47894365)

[1.1. Collaborative Filtering Process 5](#_Toc47894366)

[1.1.1. User Rating Score Data Input 5](#_Toc47894367)

[1.1.2. Formation of Neighbours 6](#_Toc47894368)

[1.1.2.1. Cosine Vector Similarity 7](#_Toc47894369)

[1.1.2.2. Pearson Correlation Coefficient 12](#_Toc47894370)

[1.1.2.3. Euclidean Distance Similarity 17](#_Toc47894371)

[1.1.3. Prediction Generation Using Score Formula 20](#_Toc47894372)

[PROGRAM 22](#_Toc47894373)

EXPLANATION OF PROGRAM 34

[OBSERVATION 54](#_Toc47894374)

[SCOPE OF FUTURE WORK 55](#_Toc47894376)

[CONCLUSION 58](#_Toc47894378)

[GLOSSARY 59](#_Toc47894379)

[REFERENCES 62](#_Toc47894381)

# INTRODUCTION

User-Based Collaborative Filtering in Recommendation Systems

A recommendation engine filters the data using different algorithms and recommends the most relevant items to users. It first captures the past behaviour of a customer and based on that, recommends products that the users might be likely to buy. Some examples of recommender systems in action include product recommendations on Amazon, Netflix suggestions for movies and TV shows in your feed, recommended videos on YouTube, music on Spotify, the Facebook newsfeed, and Google Ads.

Recommender systems use various methods to process the input data, and output recommendations to the user. The recommender systems are broadly categorized into two types: Content-based and Collaborative filtering.

Content-based recommender systems operate by comparing the description of recommendable items. This type of recommender system relies on a rich content description of items that are being recommended. Here, items may be products or services. A content-based movie recommender system will typically operate on information such as actors, directors, the genre of movies and producers, and so on. This information will be checked against the predefined preference of a user to determine the movie to be recommended to the user.

Collaborative filtering based recommender system uses a different approach. It is based on the observation that in a real-life scenario, people typically rely on friends who have similar tastes or preferences. It is built on the assumption that a possible way to determine interesting content for a user, is to find other users who have similar interest, and then recommend items that those similar users liked.

Data is collected for the Collaborative Filtering method in two types, User and Item-based. Item similarity is the most useful system for suggesting products based on how much the user would like the product. If the user is browsing or searching for a particular item, they can be shown similar items. User similarity is for checking the difference between the similarities of two users. If two users have similar preferences for a product we can assume they have similar interests.

User-Based Collaborative Filtering in Recommendation Systems

We will be using the User-based Collaborative Filtering method in our project. There are many improvements to the Recommender System and continue to be an active area of research to further enhance usability and accuracy.

# OBJECTIVE

User-Based Collaborative Filtering in Recommendation Systems

# We intend to increase our knowledge regarding the functioning of recommender systems by practicing a few techniques and basic algorithms on the dataset. This practice problem challenges the system to predict the rating for a movie given by the user by comparing ratings provided by similar users for another set of movies. This dataset is taken from the famous online MovieLens system dataset.

User-Based Collaborative Filtering in Recommendation Systems

# METHODOLOGY

## User-Based Collaborative Filtering

The user-based collaborative filtering technique is an algorithm that identifies users that are similar to the target user by locating peer users with a rating history similar to the target user.

The algorithm strives to find the user’s neighbours based on user similarities by combining the neighbour user’s rating score and then generating the neighbourhood of similar users.

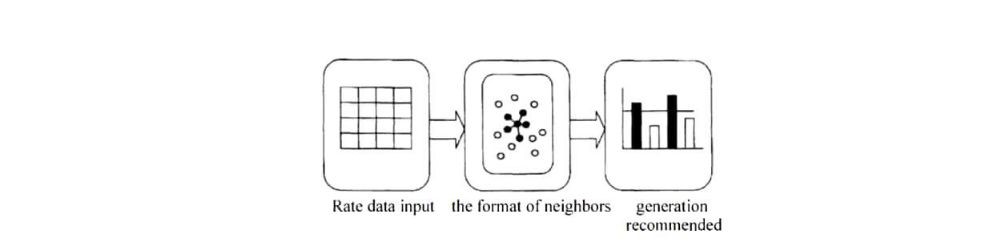
For example, let us suppose there are two users, User A and User B with similar preferences. Now, if we want to recommend Item1 to User B (target user) which has already been liked by User A, we use the UBCF technique to recommend Item1 to User B (target user). User-based collaborative filtering recommends Item1 to User B by calculating the similarity between the rating scores of two users. It uses the k-nearest neighbourhood algorithm to find the nearest users with similar preferences and then generates the prediction by calculating the predicted rating that the user might have given to that item.

### Collaborative Filtering Process

User-Based Collaborative Filtering in Recommendation Systems

The Collaborative Filtering process is divided into three main parts (as shown in Fig 1):

* Collection of user rating data matrix
* Formation of neighbours
* Prediction generation

**Fig 1: Collaborative Filtering Process**

#### User Rating Score Data Input

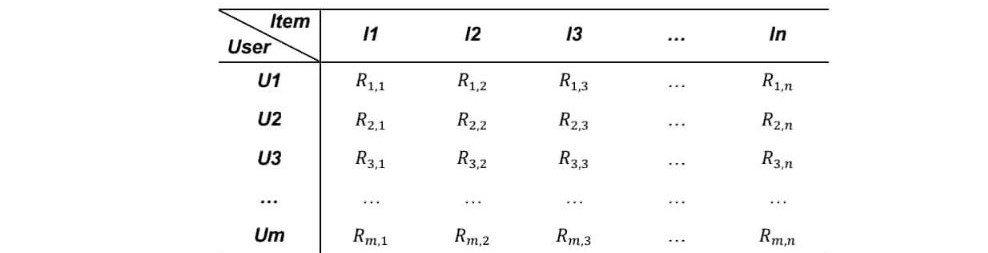
Our first step is to form a matrix of user-item ratings. Input data based on the collaborative filtering algorithm consists of the users, items, and users’ opinions on observed items in the form of *m × n* matrix (as shown in Fig 2)

Where,

*m* symbolizes the total number of users

*n* symbolizes the total number of items

*Rm,n* is the score of item *In* rated by user *Um*



User-Based Collaborative Filtering in Recommendation Systems

**Fig 2: User-Item rating matrix**

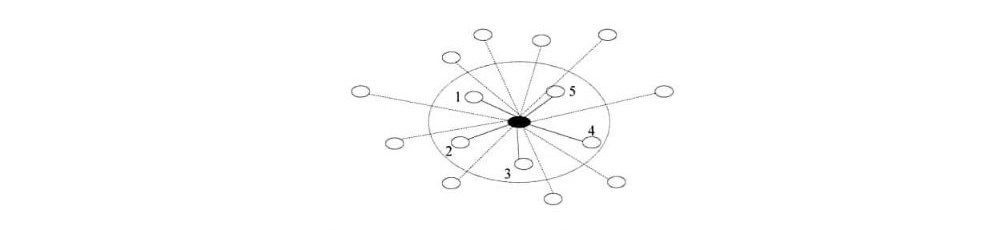
#### Formation of Neighbours

# 

In collaborative filtering, sparseness in ratings can create a hurdle in prediction generation. To solve this, a technique of neighbourhood formation of similar users is introduced. The collaborative filtering approach uses statistical techniques to analyze the similarity between users and to form a set of similar users called neighbours. A set of similarity measures is a metric of relevance between two vectors.

         The user-based similarity approach computes the relevance between users as the values of two vectors. After the similarity is calculated, it starts building neighbourhoods of the current target user.

         For example, as seen in Fig 3, the distance between the target node (black node) and every other node is calculated by a similarity measure. And then, 5 users in the middle are selected by the k-nearest neighbour algorithm where k is equal to 5.



User-Based Collaborative Filtering in Recommendation Systems

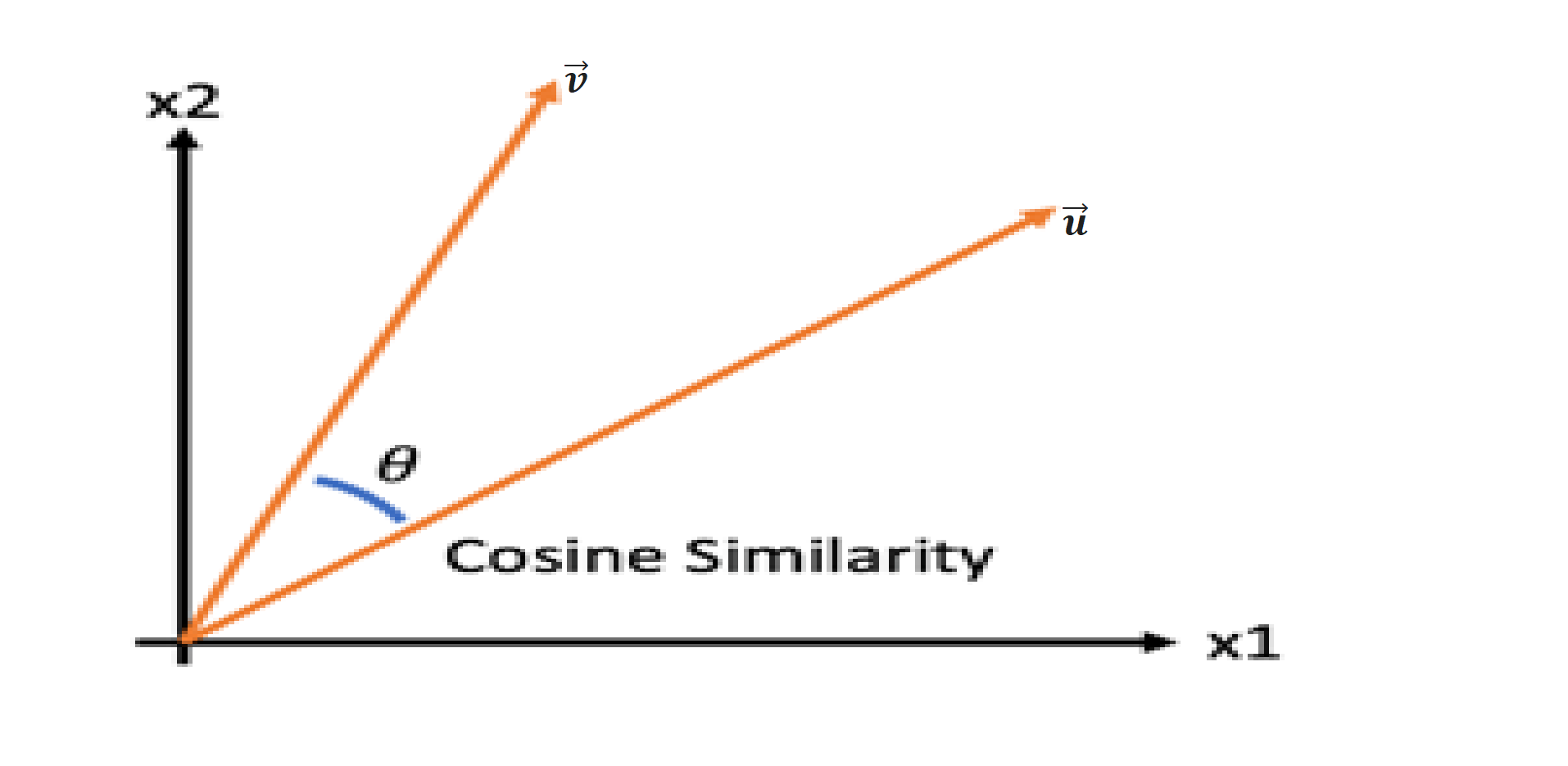
**Fig 3:** **Formation of neighbourhood**

There are several similarity algorithms that have been used in the CF recommendation algorithms for the formation of neighbourhood. In this paper, we will deal with three commonly used mathematical techniques:

* Cosine vector similarity
* Pearson correlation coefficient
* Euclidean distance similarity

##### **Cosine Vector Similarity**

Cosine Similarity is a tool used to measure how similar the concerned objects are. Mathematically, it is defined to be equal to the cosine of the angle between the two vectors. If we consider two vectors and in space then the cosine similarity between vectors and is calculated as

****

User-Based Collaborative Filtering in Recommendation Systems

**Fig 4: Cosine Similarity**

Where,

is the angle between the two vectors.

and are magnitudes of vectors and respectively.

are the components of vectors and respectively.

**In User-Based Collaborative Filtering**

It can be used on any collection of data in which the elements can be expressed in the form of vectors. The similarities between the concerned objects can be obtained by expressing them as vectors and then applying the given formula. In user-based collaborative filtering, we can compile our data in the form of data set in which the reactions of a user to different items act as the entries of the vector corresponding to that vector. In this way, all the users can be expressed in the form of vectors.

Now if we wish to find how similar two users are we just need to find the cosine similarity between their respective vectors. When the angle between the two vectors is near 0 degrees (they are in the same direction), the Cosine Similarity value is 1, meaning they are very similar. When the angle between the two vectors is near 90 degrees, the Cosine Similarity value is 0, meaning the vectors are irrelevant. When the angle between two vectors is near 180 degrees (they are in the opposite direction) Cosine similarity value is -1, meaning the vectors are very dissimilar. In the case of information retrieval using CF, the Cosine Similarity value ranges from 0 to 1. This is because the angle between two term frequency vectors cannot be greater than 90 degrees.

User-Based Collaborative Filtering in Recommendation Systems

If we wish to find the similarity of user “**a**” with user “**u**” the following formula is applied:

**=**

a: the active user (or target user)

u: another user of the system

n: the number of items that both the active user and all recommender users have rated

: the rating given by user a to item i

: the rating given by user u to item i

: the similarity between user a and user u

To calculate how similar users are, a User-Item matrix is used where the rows correspond to the users and the columns correspond to the items. Given below is an example matrix with 4 users and 5 items. The ratings are on a scale of 1-5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Item 1** | **Item 2** | **Item 3** | **Item 4** | **Item 5** |
| **User A** | **4** | **5** | **1** | **3** | **2** |
| **User B** | **3** | **2** | **3** | **2** | **4** |
| **User C** | **2** | **3** | **4** | **-** | **5** |
| **User D** | **3** | **1** | **2** | **3** | **4** |

**User-Item Matrix**

In this example,

a: the active user (User C)

User-Based Collaborative Filtering in Recommendation Systems

u: another of the user of the system (User A, User B, User D)

n: the number of items that all users have rated (Item 1, Item 2, Item 3, Item 5)

**If we calculate the similarity between User C and User A (wC,A)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | \* |  |  |
| Item 1 | 2 | 4 | 8 | 4 | 16 |
| Item 2 | 3 | 5 | 15 | 9 | 25 |
| Item 3 | 4 | 1 | 4 | 16 | 1 |
| Item 5 | 5 | 2 | 10 | 25 | 4 |
|  |  |  | Sum = 37 | Sum = 54 | Sum = 46 |

**wC,A = =** 0.7423

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | \* |  |  |
| Item 1 | 2 | 3 | 6 | 4 | 9 |
| Item 2 | 3 | 2 | 6 | 9 | 4 |
| Item 3 | 4 | 3 | 12 | 16 | 9 |
| Item 5 | 5 | 4 | 20 | 25 | 16 |
|  |  |  | Sum = 44 | Sum = 54 | Sum = 38 |

**wC,B = =** 0.9713

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | \* |  |  |
| Item 1 | 2 | 3 | 6 | 4 | 9 |
| Item 2 | 3 | 1 | 3 | 9 | 1 |
| Item 3 | 4 | 2 | 8 | 16 | 4 |
| Item 5 | 5 | 4 | 20 | 25 | 16 |
|  |  |  | Sum = 37 | Sum = 54 | Sum = 30 |

**wC,D = =** 0.9192

User-Based Collaborative Filtering in Recommendation Systems

User C has taste similar to User B and User D. User A is least similar to User C. Based on the values of and, we can say that User C is more similar to User B.

Based on the above calculation, we can predict the rating that User C would give to Item 4. The formula used for predicting the rating is:

Where,

: The predicted rating that the active user a would give to item i

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | ). |
| User A | 3 | 3 | 0 | 0.7423 | 0 |
| User B | 2 | 3 | -1 | 0.9713 | -0.97 |
| User D | 3 | 2.5 | 0.5 | 0.9192 | 0.46 |
|  |  |  |  | **Sum = 2.6328** | **Sum = -0.5117** |

Here,

: Rating given by user u (User A, User B, User D) to Item 4

Therefore, the rating that user C would give to item 4 is:

**= 3.3056**

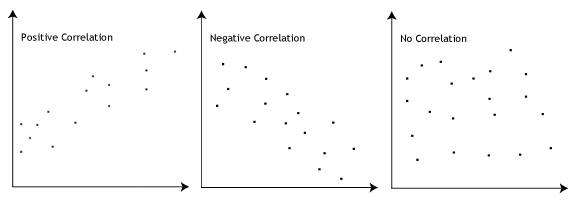
##### **Pearson Correlation Coefficient**

User-Based Collaborative Filtering in Recommendation Systems

The Pearson Product-Moment Correlation Coefficient or simply, the Pearson Correlation Coefficient is a measure of the strength of a linear association between two variables. In other words, a Pearson Product-Moment Correlation attempts to draw a line of best fit through the data of the two variables and the Pearson Correlation Coefficient ‘r’ indicates how far away all these data points are from the line of best fit.

The value of the Pearson Correlation Coefficient ‘r’ lies in the range of -1 to +1.

* If the value of ‘r’ is 0, it indicates that there is no association between the two variables.
* If ‘r’ is greater than 0, it indicates that there is a positive association between the two variables, that is, as the value of one variable increases, so does the value of the other variable.
* If ‘r’ is less than 0, it indicates that there is a negative association between the two variables, that is, as the value of one variable increases, the value of the other variable decreases.

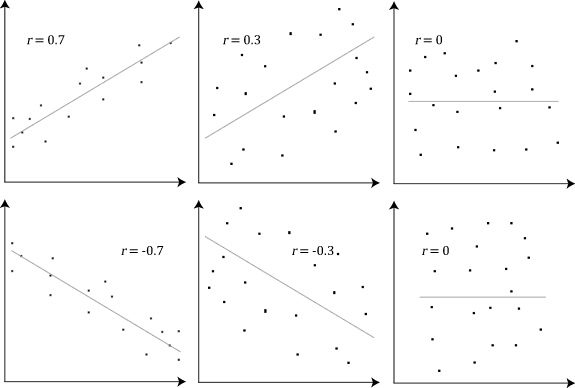


**Fig 5: Positive, Negative and No Correlation**

The stronger the linear association between the two variables, the closer will be the values of r to +1 or -1, depending on whether the relationship is positive or negative. When r = ±1, there is said to be a perfect correlation with all the data points being in a perfectly straight line. The closer the value of r to 0, the greater is the variation around the line of best fit.

User-Based Collaborative Filtering in Recommendation Systems

|  |  |  |
| --- | --- | --- |
| **Strength of Association** | **Pearson Correlation Coefficient r** | |
| **Positive** | **Negative** |
| **Weak**  **Moderate**  **Strong** | **0.1 to 0.3**  **0.3 to 0.5**  **0.5 to 1.0** | **-0.1 to -0.3**  **-0.3 to -0.5**  **-0.5 to -1.0** |



**Fig 6: Strength of Association**

The Pearson Correlation has the following assumptions:

* The two variables should be continuous, that is, the scale of measurement should be in an interval or a ratio.
* There should be paired observations, that is, the data points must be in pairs.

User-Based Collaborative Filtering in Recommendation Systems

* There should be independence of cases.
* There should be a linear relationship between the two variables.
* The two continuous variables should be normally distributed.
* There should be homoscedasticity, that is, the variances along the line of best fit should remain similar as we move along the line.
* There should be no univariate or multivariate outlier.

**In User-Based Collaborative Filtering**

Pearson Correlation Coefficient is used in User-Based Collaborative Filtering as a similarity measure to compute the similarity between users, based on historical rating information. When the value of the Pearson Correlation Coefficient is closer to +1, it implies that the two users have similar tastes, whereas when the value is closer to -1, it implies that the users have divergent tastes.

The formula for calculating the Pearson Correlation Coefficient is:

**=**

Where,

a : the active user (or target user)

u: another of the user of the system

n: the number of items that both the active user and all recommender users have rated

: the rating given by user a to item i

: the rating given by user u to item i

: the average rating of the active user a

User-Based Collaborative Filtering in Recommendation Systems

: the average rating of the other user u

: the similarity between user a and user u

To calculate how similar users are, a User-Item matrix is used where the rows correspond to the users and the columns correspond to the items. Given below is an example matrix with 4 users and 5 items. The ratings are on a scale of 1-5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Item 1** | **Item 2** | **Item 3** | **Item 4** | **Item 5** |
| **User A** | **4** | **5** | **1** | **3** | **2** |
| **User B** | **3** | **2** | **3** | **2** | **4** |
| **User C** | **2** | **3** | **4** | **-** | **5** |
| **User D** | **3** | **1** | **2** | **3** | **4** |

**User-Item Matrix**

In this example,

a: the active user (User C)

u: another of the user of the system (User A, User B, User D)

n: the number of items that all users have rated (Item 1, Item 2, Item 3, Item 5)

: the average of the active user’s (User C) ratings, which is = 3.5

: the average of user A’s ratings, which is = 3

: the average of user B’s ratings, which is = 3

: the average of user D’s ratings, which is = 2.5

Calculation of Correlation between user C and user A:

User-Based Collaborative Filtering in Recommendation Systems

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | - | - | (- )2 | (- )2 | (- ).(- ) |
| 2 | 4 | -1.5 | 1 | 2.25 | 1 | -1.5 |
| 3 | 5 | -0.5 | 2 | 0.25 | 4 | -1.0 |
| 4 | 1 | 0.5 | -2 | 0.25 | 4 | -1.0 |
| 5 | 2 | 1.5 | -1 | 2.25 | 1 | -1.5 |
|  |  |  |  | **Sum = 5** | **Sum = 10** | **Sum = -5** |

**wC,A = = -0.7071**

Similarly,

The correlation between User C and User B is:

**wC,B = = 0.6324**

The correlation between User C and User D is:

**wC,D = = 0.4**

Thus, in the above example we can say that User C and User A have divergent taste. User C has taste similar to User B and User D. Based on the values of WC,B and WC,D, we can say that User C is more similar to User B.

Based on the above calculation, we can predict the rating that User C would give to Item 4. The formula used for predicting the rating is:

Where,

: the predicted rating that the active user a would give to item i

|  |  |  |  |
| --- | --- | --- | --- |
| User-Based Collaborative Filtering in Recommendation Systems |  |  | ). |
| 3 | -0.7071 | 0 | 0 |
| 2 | 0.6324 | -1 | -0.6324 |
| 3 | 0.4 | 0.5 | 0.2 |
|  | **Sum = 0.3253** |  | **Sum = -0.4324** |

Here, is the rating given by user u (User A, User B, User D) to Item 4

Therefore, the rating that user C would give to item 4 is:

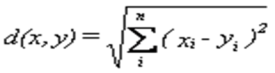
**= 2.1707**

##### **Euclidean Distance Similarity**

Euclidean Distance is considered the traditional metric for problems of geometry. It is one of the most used algorithms in cluster analysis. Clustering consists of grouping certain objects that are similar to each other. It is just the ordinary distance that we calculate between two points in the geometry. Euclidean Distance between two points, x and y is the length of the line segment connecting x and y.

It is defined as the square root of the sum of squared differences between corresponding elements of 2 vectors.

https://lh3.googleusercontent.com/84B9gODAtJIcT6vT6dlS6-8cfER7USZkyyfNwNxavJSpuxinHaxDIdAbw9Fh2OFYC971OHzVrTChDyFY_DVG2a-mBOhjeu0ag7vmZyiaQkjzBCPYknB4SIGLRiUna0K1D56lVckD



Where,

and are the rating score of an item given by two different users for the same item

n is the number of commonly rated items.

This formula is derived from the Pythagoras Theorem. Euclidian Distance is only appropriate for data measured on the same scale.

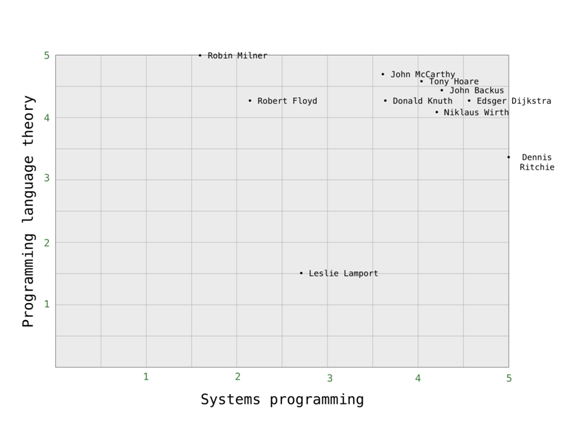
User-Based Collaborative Filtering in Recommendation Systems

The Euclidean distance tool is used frequently as a stand-alone tool for applications, such as finding the nearest hospital for an emergency helicopter flight. Alternatively, this tool can be used when creating a suitability map, when data representing the distance from a certain object is needed.

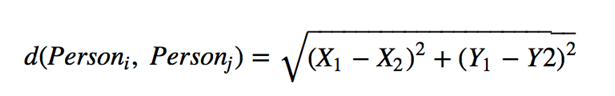
The greater the distance between two points, the lower is the similarity between them. It forms coordinates to put preference values between items and measures Euclidean distance between each point. When the distance value between two points sim(i, j) is large, it means that those points are not similar.

**Example 1:**

Plot shows the users considering their tastes on both systems programming and programming language theory. It shows the people that have ranked both items in a preference space defined by those items, and the scores given by people to each item independently.



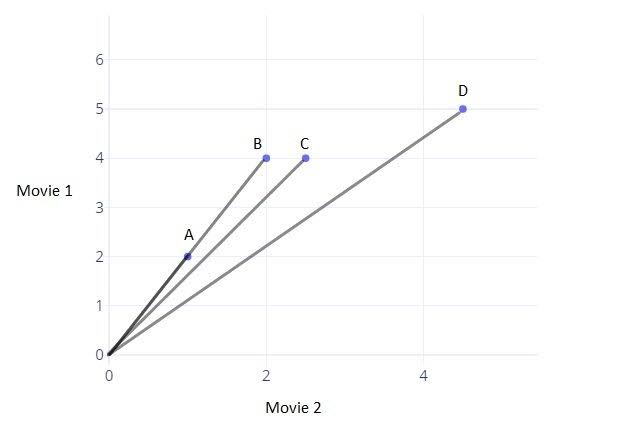
**Fig 7**



**Example 2:**

User-Based Collaborative Filtering in Recommendation Systems

This is a plot showing the ratings of 4 users for movie 1 and movie 2.



**Fig 8**

User A [1,2] ; User B [2,4] ; User C [2.5,4] ; User D[4.5,5]

Euclidean Distance (User A, User B) =

=

= = = 2.23

Euclidean distance (User A, User C) =

=

=

= = = 2.5

Therefore, Euclidean Distance (User A, User C) is greater than Euclidean Distance (User A, User B)

User-Based Collaborative Filtering in Recommendation Systems

2.5 > 2.23

Thus, User A and User B are more similar than User A and User C.

#### Prediction Generation Using Score Formula

Once CF computes the similarity between users (in UBCF) and then finds the set of most similar user or similar items, it generates prediction of the target user’s interest as the most significant step in CF.

**Prediction Computation of UBCF**

Since UBCF gets the neighborhood of user, UBCF can calculate the predictive rating for the target User ‘*’* on the target Item ‘*’.* It is scaled by the weighted average of all neighbors’ ratings on the target Item *i* by using the following formula:

***) =***

Where a is the target user and ‘’ is the item for which the User ’s prediction is to be calculated and **)**is the predicted score.

is the average rating of the target User *u.)*

User-Based Collaborative Filtering in Recommendation Systems

is the average rating of the neighbor User v.

is the rating of the neighbor User u to the target item *i*.

U is the set of similar neighbors.

wa,u is the similarity (weight) of the target User *a* and the neighbor User u*,* which is calculated using the following three methods:

1. Cosine Vector Similarity
2. Pearson Correlation
3. Euclidean Distance

In this case, our predicted score is equal to the sum of the ratings that each user gave to that item subtracting the average rating of that user multiplied with some similarity (weight) which is of how much this user is similar or supposed to contribute to the predictions of another user. This is the similarity between user ‘*’* and user ‘*’*.

We subtract the average ratings from each user's rating and we use the weighted average, that is, the similarity instead of the simple mean because the problem is with the types of users we are handling. It starts with the fact that people rate often on very different scales. We can increase the efficiency of this algorithm if we normalize the user’s rating. One way to do that is to compute s(,*’*), that is, the predicted score as the average rating that user gives to each item plus some deviation and the deviation is going to be how much this item is better or worse than average.

User-Based Collaborative Filtering in Recommendation Systems

# PROGRAM

# OBSERVATION

User-Based Collaborative Filtering in Recommendation Systems

# Taking a hypothetical case of comparison where if a distribution of ranking from user A tends to be higher than user B i.e. User B is inclined to rate more critically, Euclidean Distance and Cosine Similarity would classify them as dissimilar without regarding the correlation between the two users. There can still be a near-to perfect similarity if the differences between their rankings are consistent. If our system is limited to just using Euclidean Distance and Cosine similarity, this might give an inaccurate result because user B is consistently more critical than User A. Since the Pearson Correlation Coefficient is a measure of the strength of a linear association between two variables, it automatically detects and accommodates the weight of harshness between two users. Hence, it is better to use the Pearson Correlation Coefficient in cases like these.

# SCOPE OF FUTURE WORK

User-Based Collaborative Filtering in Recommendation Systems

Research in this field looks saturated from the outside yet there are a lot of areas to work on. The next generation of recommendation systems may include the following improvements:

* ***Cross-Domain Recommendation***

There is a need for a unified model of preference for an individual that explains how different domains interact and inform our preferences. For example, most recommender systems work on only one attribute, movies, music genres, etc. An algorithm that works across these different attributes can open up a lot of opportunities for service providing platforms.

* ***Context-aware Recommendation***

Fine-grained information like live location and real-time updates from mobile phones can be a gold mine for recommendation systems. Not a lot of research has been done in this aspect, which can elevate the statuses of these systems.

* ***Recommendation and Privacy***

The usage of private information of innumerable users by these recommender systems can prove to be hazardous if adequate security isn’t taken. Privacy in the recommendation is a major concern, and it would be great to see some theoretical and empirical work.

* ***Recommendation and Social Networks***

One major problem arises in designing a recommender system in the social context. In the times of multiple social media and people’s dependency on it increasing day by day, the product/service/information they’re provided depends on their network’s preferences. Social networks themselves act as vehicles of information, diffusing ideas and information across the network. This means that user preferences in a network are not only dynamic, but they also have an interdependency with the network.

* ***New Domains for Recommendation***

A lot of work can be done in new domains where the concept of using recommendation systems hasn’t yet been used. Using algorithms in discussions and comments in a forum like Reddit, the expert recommendation in domains like LinkedIn, recommending videos using more than text/tag features can prove to be interesting.

User-Based Collaborative Filtering in Recommendation Systems

* ***More Relevant Recommendations***

By digging deeper into customers’ interests and preferences, recommendation systems will be able to present users with more-relevant, predictive recommendations.

* ***Incorporate Item Profitability***

Instead of having recommendations based solely on a customer’s browsing history and past purchases, this would allow businesses to control how much a profit-based recommendation differs from the traditional recommendation and to set a balance so that customer trust would not be compromised.

* ***Increase Product Reach***

Each retailer has an individual catalog of products, improved recommendation systems would be able to access a broader range of merchandise to include new or niche items in shoppers’ recommendations.

* ***Reach shoppers through multiple channels***

Next-generation recommendation systems should be able to reach customers across a range of channels including email, social media, on off-site shopping widgets, mobile apps, and the retail customer service centers.

***Scope of Recommendation system in e-Commerce***

* ***Increase Average Order Value***

Average order values typically go up when a *recommendation engine*in uses to display personalized options. Advanced metrics and reporting can definitively show the effectiveness of a campaign.

* ***Increase Number of Items per Order***

 In addition to the average order value rising, the number of items per order also typically rises when a **recommendation engine** is employed. When the customer is shown options that meet his interest, he is more likely to add items to his purchase.

* ***Control Merchandising and Inventory Rules***

 A recommendation engine can add marketing and inventory control directives to the customer’s profile to feature products that are promotional prices, on clearance or overstocked. It gives the flexibility to control what items are highlighted by the *recommendation system*

User-Based Collaborative Filtering in Recommendation Systems

* ***Deliver Relevant Content***

By analysing the customer’s current site usage and his previous browsing history, a **recommendation engine**can deliver relevant product recommendations as he shops. The data is collected in real-time so the software can react as his shopping habits change.

* ***Engage Shoppers***

Shoppers become more engaged in the site when personalized product recommendations are made. They can delve more deeply into the product line without having to perform search after search.

# 

User-Based Collaborative Filtering in Recommendation Systems

# CONCLUSION

Recommender Systems have great value in recommending relevant resources to users. They can be quite useful in finding novel and serendipitous recommendations. The effectiveness of a recommender system depends on the algorithm it uses to find interesting resources.

This Project presents a web-based movie recommendation system written in Python Programming Language. The main aim of this project was to find the mathematics involved behind a general recommendation system and also to analyze the program behind it. The system involved is The Movie Recommendation System using the MovieLens dataset. It plays a significant role in identifying a set of movies for users based on user interest. Recommender Systems work on the strategy that focuses on dealing with user's interests and based on their previous reviews, movies are recommended to them.

In this project, we have analyzed the algorithm behind User-Based Collaborative Filtering used in Recommender Systems. Collaborative Filtering only needs the user's historical preference on a set of items. User-Based Collaborative Filtering is easy to implement, context-independent and more accurate as compared to others like Content-Based Collaborative Filtering.

# GLOSSARY

User-Based Collaborative Filtering in Recommendation Systems

**IMPORT:** Import is a keyword in python which is used to import/install required libraries in the local scope.

Example - from sklearn.metrics.pairwise import cosine\_similarity as cs

                  from and as are also keywords.

**LIBRARIES:** Python Library is a collection of core modules that contains reusable chunks of codes that can be used in other programs/projects.

 Important Libraries that we used are Pandas, Numpy, Matplotlib, etc.

**FUNCTIONS:** Python functions are blocks of code that are used to solve a problem after doing the specific calculation. These are reusable codes, which runs only when we call the function. There are two types of functions: inbuilt and user-defined.

**INBUILT FUNCTIONS:** Inbuilt functions are those functions which are already defined in python. End-users can use it directly by calling its name.

 Example - len(), merge(), etc.

**Inbuilt Functions that we used :**

1.**head():** head() is used to get first n rows, by default it gives the first five rows if no parameter passed in the head function.

2. **describe() :** describe() in pandas library is used to compute summary of numeric columns in dataframe.

3.**groupby() :** groupby() is used to split data in groups based on some criteria.

Example - dataframe.groupby(by=’userid’), here userid is a criterion based on which data splits.

4.**mean():** mean() is used to compute the mean/average of a given list of data.

5. **hist():** hist() function in matplotlib library is used to plot histograms.

6. **title():** title() function in matplotlib library is used to specify title of visualization depicted.

7. **xlabel():** xlabel() function in the pyplot module of matplotlib library is used to set label for the x-axis.

8.**merge():** merge() function is used to join columns from two dataframes.

9. **rename():** pandas rename() function is used to rename any index, column or row.

10.**pivot\_table():** pivot\_table() creates spreadsheet-style pivot table as dataframe.

User-Based Collaborative Filtering in Recommendation Systems

11. **apply():** apply() function is used to apply/pass a function along an axis of the dataframe.

12.**Lambda :** lambda() is an anonymous function. It is a keyword that is used to create anonymous functions. Its expression - lambda arguments: expression

13. **fillna():** fillna() is used to fill or replace NA/NAN values using the given method.

14.**cosine\_similarity():** cosine\_similarity() is used to compute the similarity between two samples, samples can be any number of lists, items, etc.

15. **fill\_diagonal() :** fill\_diagonal() allows users to fill diagonal values with the values given as parameters.

16.**DataFrame() :** DataFrame() function in pandas is used to create/convert a dataframe from dict, matrix.

19. **argsort():** argsort() in numpy library is used to perform an indirect sort along the given axis.

20. **sort\_values() :** pandas sort\_values() function is used to sort a dataframe in ascending or descending order or passed column.

21. **format() :** format() function is used to format stings.

22. **astype():** astype() is used to cast a pandas object to a specified data type. This function is also used to convert existing columns to categorical type.

23. **join():** join() function is used in string, it returns a string in which elements are separated by string separator.

24. **notna():** python notna() function detects the existing values in dataframe and returns a boolean object having the same size as the object, on which it is applied.

25. **any():** python any() is used to detect if any of the items in an iterable object is true or not. It returns true if any of the item is true.

26. **tolist():** tolist() function returns a list of the values.

27.**isin():** isin() function is used to check each elements in the dataframes.

28. **squeeze():**squeeze function is used to remove single-dimensional values from the shape of an array.

29. **split():**split() function splits a string into a list, a separator can also be specified to separate elements.

30. **map():** map() function is used to apply a function to all the elements of specified iterable and returns map object.

31. **set():** python set() is used to convert an iterable into a set datatype which is a sequence of iterable elements with distinct elements.

32. **list():** python list() is used to convert a sequence into a list datatype.

33. **concat():** pandas concat() is used to concat pandas object along with a particular axis and optional set logic along the other axis.

34. **append():** append() in python is used to append/add an element at the end of the list. Also, in the case of dataframes, append() is used to append rows of other dataframe to the end of a given dataframe, hence return a new dataframe.

User-Based Collaborative Filtering in Recommendation Systems

35. **print():** print() is used to print a specified message and output on the screen.

36. **filterwarnings():** filterwarnings() is used to avoid/remove warnings to get print on the screen.

37.**notnull():** pandas notnull() is used to check and manage null values in dataframe.

38. **sum():** sum() function is used to sums up all given numbers.

39. **read\_csv():**read\_csv is a function in pandas which is used to read csv(comma-separated files) and to do operations on it.

Syntax: pd.read\_csv(“File Location”)

**USER DEFINED FUNCTIONS:** Users can also create their functions to solve a problem , these functions are known as user-defined functions. We can define functions using a def keyword.

 Example - def k\_nearest\_neighbours(df,k):

 Under this statement, we can write a statement to perform calculations or to solve problems.

**Loops:** Loops in python iterate over the items of any sequence such as list or string. There are two types of loops:

1. For loop

2. While loop

In our program, we have used only for loop.

**For Loop:** for loop is used for a fixed number of iteration.

Syntax of for loop- for i in ‘sequence’:

                                where for, in are keywords and i is a variable.

**iloc and loc:**

In the pandas library, there are many ways to filter out data. Two of them are:

Loc and Iloc

**loc:**loc is label based, implies to filter out data from a given data frame. We have to specify the name of rows and columns that we need to filter out.

Syntax: pd.dataframe.loc[]

**iloc:** iloc is integer index-based, implies to specify rows and columns by their integer index.

Syntax: pd.dataframe.iloc[]

# 

User-Based Collaborative Filtering in Recommendation Systems

# REFERENCES

* Dharaneeshwaran, & Nithya, S & Srinivasan, A & Mohan, Senthilkumar. (2017). Calculating the user-item similarity using Pearson's and cosine correlation. 1000-1004. 10.1109/ICOEI.2017.8300858.
* Lee, Yunkyoung, "RECOMMENDATION SYSTEM USING COLLABORATIVE FILTERING" (2015). Master's Projects. 439.

DOI: <https://doi.org/10.31979/etd.5c62-ve53>

<https://scholarworks.sjsu.edu/etd_projects/439>

* <https://builtin.com/data-science/recommender-systems>
* <https://medium.com/ai-society/a-concise-recommender-systems-tutorial-fa40d5a9c0fa>
* <https://realpython.com/build-recommendation-engine-collaborative-filtering/>
* <https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f>
* <https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php>
* <https://www.toptal.com/algorithms/predicting-likes-inside-a-simple-recommendation-engine>
* <https://www.kdnuggets.com/2017/08/recommendation-system-algorithms-overview.html>
* <https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluation-and-cold-start-6f696683d0ed>
* <https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf>
* <https://medium.com/sfu-cspmp/recommendation-systems-user-based-collaborative-filtering-using-n-nearest-neighbors-bf7361dc24e0>
* <https://grouplens.org/datasets/movielens/>
* <https://github.com/ashaypathak/Recommendation-system/blob/master/Movie_Recommendation.ipynb>
* <https://programmer.ink/think/movie-recommendation-using-surprise-package.html>

User-Based Collaborative Filtering in Recommendation Systems

* [https://learning.tcsionhub.in//EForms/loginAction.do?subAction=ViewLoginPage&orgId=1016&formId=2151#1%40IID%40Home%2FCatalogue%3DredirectTo~%40~%26subAction%3DviewMicrosite%26Link%3Dhttps%3A%2F%2Fcdn4.tcsion.com%2Fdotcom%2FICMP%2Flayout%2FMicrosite%2FNPTEL%2FData\_Structures\_and\_Algorithm.html%26id%3DNPTELCSDSA12042017%26name%3DData%2BStructures%2Band%2BAlgorithms](https://learning.tcsionhub.in/EForms/loginAction.do?subAction=ViewLoginPage&orgId=1016&formId=2151#1%40IID%40Home%2FCatalogue%3DredirectTo~%40~%26subAction%3DviewMicrosite%26Link%3Dhttps%3A%2F%2Fcdn4.tcsion.com%2Fdotcom%2FICMP%2Flayout%2FMicrosite%2FNPTEL%2FData_Structures_and_Algorithm.html%26id%3DNPTELCSDSA12042017%26name%3DData%2BStructures%2Band%2B)
* [https://www.google.com/url?sa=t&source=web&rct=j&url=https://sg.inflibnet.ac.in/jspui/bitstream/10603/62351/9/chpt6.pdf&ved=2ahUKEwiwiIH-hfLqAhXf83MBHbPgAioQFjARegQIAhAJ&usg=AOvVaw0tp3isESWygvUGl9\_zAvUS&cshid=1596011490911](https://sg.inflibnet.ac.in/jspui/bitstream/10603/62351/9/chpt6.pdf)
* [https://www.google.com/url?sa=t&source=web&rct=j&url=http://ijcst.com/vol53/2/pallab-dutta.pdf&ved=2ahUKEwiwiIH-hfLqAhXf83MBHbPgAioQFjASegQIAxAB&usg=AOvVaw2qNnX6R8TZ1nB\_ltItleUB&cshid=1596011490911](http://ijcst.com/vol53/2/pallab-dutta.pdf)
* <https://towardsdatascience.com/recommender-system-a1e4595fc0f0>
* <https://emerj.com/ai-sector-overviews/use-cases-recommendation-systems/>