International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3s
DOI: https://doi.org/10.17762/ijritcc.v11i3s.6180
***Article Received**: 16 December 2022 **Revised**: 24 January 2023 **Accepted**: 30 January 2023*

A Comparative Analysis of Collaborative Filtering Similarity Measurements for Recommendation Systems

[1]Praveen Kumar, [2]Dr. Mukesh Kumar Gupta, [3]Dr. Channapragada Rama Seshagiri Rao, [4]M Bhavsingh, [5]M.Srilakshmi
[1]Research Scholar, Department of CSE, Suresh Gyan Vihar University, Jaipur.
[2]Associate Dean Research, Suresh Gyan Vihar University, Jaipur.
[3]Department of CSE, Professor & Principal, Vignana Bharathi Engineering College, Chintapalliguda(v).Ibrahimpatnam
Email: crsgrao@gmail.com
[4]Assistant Professor, Department of CSE, Ashoka Women's Engineering College,Kurnool, Andhra Pradesh
Email: bhavsinghit@gmail.com
[5]Associate Professor, Department of CSE, G. Pullaiah College of Engineering and Technology (Autonomous), Kurnool.
Email: marri_srilakshmi67@yahoo.co.in

Abstract: Collaborative Filtering (CF) is a widely used technique in recommendation systems to suggest items to users based on their previous interactions with the system. CF involves finding correlations between the preferences of different users and using those correlations to provide recommendations. This technique can be divided into user-based and item-based CF, both of which utilize similarity metrics to generate recommendations. Content-based filtering is another commonly used recommendation technique that analyzes the attributes of items to suggest similar items. To enhance the accuracy of recommendation systems, hybrid algorithms that combine CF and content-based filtering techniques have been developed. These hybrid systems leverage the strengths of both approaches to provide more accurate and personalized recommendations. In conclusion, collaborative filtering is an essential technique in recommendation systems, and the use of various similarity metrics and hybrid techniques can enhance the quality of recommendations.

Keywords: Recommendation System; Collaborative Filtering; Similarity Measurements; Model-Based; Memory-Based; Filtering Systems;

I. INTRODUCTION

Nowadays, with the increasing popularity of the internet, people use smartphones, tablet PCs, and other intelligent devices that rapidly generate a massive volume of digital data . As a result, several e-applications have entered the age of the information overload problem. Due to this, the recommender system plays a critical role in dealing with the information overload problem. Recommendation systems (RSs) are an intelligent computer-based technique that anticipates and assists people in selecting products from a large pool of items based on their adoption and usage . The RS has improved significantly over time to make the process of looking for or searching for items easier for the user. Multinational corporations rely heavily on the efficacy of their product suggestion system .

RSs are integral to the operations of many successful businesses, like Flipkart, Amazon, eBay, Netflix, MovieLens, IMDb, etc. Most people who have used the internet have probably used an RS at some point . For instance, Facebook suggests friends, YouTube and Tiktok suggest movies that go well together, Glassdoor suggests jobs that are a good fit, TripAdvisor suggests vacation spots, Goodreads suggests books that would be of interest, and so on. Amazon, Netflix, LinkedIn, and Pandora all employ RSs to help customers find new and interesting content, which improves the user experience and generates additional income. Websites for online shopping are just one area where RSs have proven useful; others include e-government, in which the government uses the internet to provide services to citizens; e-learning, in which the entire educational process is carried out online with the aid of electronic devices, e-tourism, and e-resources, among others .

Various filtering approaches have been applied in the literature to make an effective and enhanced RS . Some of the main filtering approaches are Content-based filtering systems (CBFS), Collaborative Filtering (CF), Knowledge-based Filtering (KBF), Filtering that takes into account the surrounding information and context, or CAF . CF is well recognized as a powerful tool for RS researchers. CF searches a vast population to locate a subset of individuals who share the same preferences as a given user. The two most common methods in CF-based RS are model-based and neighbourhood-based or memory-based. Model-based CF algorithms first construct a model of users' ratings to provide product recommendations . As a result, the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3s
DOI: https://doi.org/10.17762/ijritcc.v11i3s.6180
***Article Received**: 16 December 2022 **Revised**: 24 January 2023 **Accepted**: 30 January 2023*

algorithm adopts a probabilistic viewpoint, seeing the CF procedure as an exercise in determining the anticipated value of a user prediction. Model-based approaches efficiently address the problem of sparsity by reducing a sparse dataset to a low-dimensional matrix. As a result of this, the system's scalability is improved. However, reducing the dataset size reduces the quality of the predictions, resulting in inaccurate estimates. Model-based approaches fail in a highly dynamic dataset since they are time-consuming. The repeatedly retraining model techniques are likewise not scalable .

Memory-based CF uses user ratings to determine how similar users or products are. Memory-based CF calculates the similarity between people or things based on their evaluations . Similar users or items are then grouped in rating prediction approaches. Memory-based approaches provide far more accurate and defensible predictions when compared to model-based approaches. Furthermore, memory-based CF can be classified into user and item-based approaches. In a user-based approach, users recommend items based on the similarity computed between every possible pair of users in the dataset .

Similarly, in an item-based approach, items are recommended to users based on the similarity computed between every possible pair of items in the dataset. The item-based approach is highly preferred over the user-based approach since the predicted ratings tend to be much more consistent . However, in reality, the datasets used in many business systems are very sparse, as not all users generally rate. In such a scenario, many commonly used similarity measures have several issues. P. K. Singh et al. have used items with categorical attributes with rating values of users to minimize the sparsity. Additional parameters in similarity measures necessitate some optimization strategy, but P. K. Singh et al. have not specified any optimization technique .

The preceding observations indicate the areas of comparison in a user-based CF. The following philosophical contribution has been made in this paper to help define the issues mentioned earlier, and research outperforms existing methods in those scenarios to assess their pros and cons. The paper is further structured as follows; section 2 discusses the background of the RS and some significant/breakthrough works of CF-based RS. Section 3 comprises the methods and results of the paper with various scenario-based implementations. Section 4 concludes the paper with a significant comparative analysis, research findings & the future scope.

II. BACKGROUND

A literature review of recommendation systems reveals that they are efficient tools for filtering online information. They are used to filter and retrieve data and have become an active area of research . There are various recommendation systems based on filtering techniques, such as case-based, utility-based, and computational intelligence-based. Recommendation systems have been applied in cybersecurity contexts to defend against malware and have been employed in diverse applications such as e-commerce, market basket data, and online information retrieval . According to a review of their implementation, there is no universally accepted set of criteria or techniques for evaluating the performance of state-of-the-art recommender systems.

Collaborative Filtering, content-based Filtering, hybrid approaches, session-based recommender systems, reinforcement learning for recommender systems, multi-criteria recommender systems, and risk-aware recommender systems are some examples of the algorithms that can be used in recommendation systems . The concept of collaborative Filtering rests on the presumption that those who have shown agreement in the past will continue to agree in the future . It works by analyzing user interactions with items to generate recommendations. Methods of Filtering based on content use a profile of the user's preferences in addition to a description of the item being filtered . The collaborative Filtering and the content-based filtering methods are combined in hybrid techniques. Session-based recommender systems use information about a user's current context to generate recommendations . Reinforcement learning for recommender systems uses deep learning techniques to combine collaborative Filtering and content-based models . Multi-criteria recommender systems consider multiple criteria when generating recommendations Risk-aware recommender systems consider potential risks when making recommendations Memory-based collaborative recommendation engine algorithms can generate strong recommendations based on user interactions with items .

To create recommendation systems that improve their service to consumers over time, researchers have turned to a family of algorithms known as collaborative Filtering. It finds similarities between users and items, allowing for serendipitous recommendations. Memory-based and model-based collaborative filtering algorithms are the two most common kinds. One may talk about user-item Filtering and item-item Filtering in terms of memory-based algorithms . Item-item filtering considers users' ratings similar to those in question to locate products they liked Model-based algorithms use low-dimensional factor models to predict user ratings for an item. Examples of model-based algorithms include matrix factorization, TransE, and hand-engineered embeddings .

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3s
DOI: https://doi.org/10.17762/ijritcc.v11i3s.6180
***Article Received**: 16 December 2022 **Revised**: 24 January 2023 **Accepted**: 30 January 2023*

Collaborative Filtering is a popular technique used in recommender systems to determine the set of users with similar behaviour regarding selected items. To do this, similarity measures are used to compare user ratings and item characteristics. Cosine similarity is often used for item-based collaborative Filtering, while Pearson correlation coefficient and adjusted cosine similarity are more commonly used for user-based collaborative Filtering. The appropriate similarity measure selection depends on the data analysis type and the desired outcome .

One of the most important features of neighbourhood-based models in Collaborative Filtering is the similarity estimation between two users or things. A similarity measure or function is a real-valued function used in statistics and related fields that quantifies the similarity between two objects . This enables us to group similar items or users, which results in more accurate recommendations. Historically, various metrics have been used in Collaborative Filtering-based recommendation systems. They have primarily been used to determine the proximity of users or items. It is a statistical computation that determines the similarity of two objects. Cosine similarity, Adjusted Cosine similarity, Euclidean Distance, Pearson Correlation, Spearman Correlation, Manhattan Distance, Chebychev Distance, and Jaccard similarity are just a few of the traditional similarity measures that are frequently used .

III. METHODS & RESULTS

To determine the degree of similarity between a pair of products or people, recommendation systems often employ several different similarity metrics . The most popular are Euclidean distance similarity and Cosine similarity . A pair of points' Euclidean distance is the shortest possible path between them, whereas a pair of vectors' Cosine similarity is the smallest possible angle between them . When combined, these metrics form a similarity matrix that may be used to match a user with services and goods that they are likely to enjoy . For the movie example, if a user enjoys a specific film, other users who have rated that film similarly will be suggested as possible viewers . Similarly, if one user has given positive ratings for certain movies, then other movies with similar characteristics will be recommended to that user [1].In addition to these measures, other methods, such as the Pearson correlation coefficient and the Jaccard index, can be used in recommendation systems . These methods can be used to compare data types, such as numerical and categorical data. They can also compare different types of cold and active users . Various existing SMs are not provided with the optimal similarity value between users; as a result, the accuracy of CF-based RS does not attain its high level. The condition, as mentioned earlier, opens a future scope of modifications in the similarity computation methods.

However, while Collaborative Filtering User Behaviour (CFUB) mitigates some of the shortcomings of commonly used SMs, an improved item-based collaborative filtering, it also fails in particular user rating patterns. BC has various limitations, and because CFUB uses BC to compute user and item similarity, CFUB may suffer certain limitations . For example: Suppose three customers rate six products on a scale of 1-5. Then, the rating patterns of these customers become C1={2,2,0,0,0,0}, C2={0,0,5,0,1,0}, and C3={0,0,0,4,0,3}. Here, 0 identifies that a customer does not rate a particular product. On this rating pattern of customers, similarity using CFUB cannot compute the similarity value of customers and products, as illustrated in table 1.

Table 1: Failures of Customer Behaviour

Customer C1 on target product P1 = Action, Adventure	**Customer C1 on target product P2 = Action, Adventure, Animation**	**Customer C1 on target product P3 = Children's movies, Documentary**	**Customer C1 on target product P4 = Crime, Drama, Thriller**	**Customer C1 on target product P5 = Romance**	**Customer C1 on target product P6 = Fantasy**
$\mathrm{Sim}^{C1}(P1, P3) = 0$	$\mathrm{Sim}^{C1}(P2, P3) = 0$	$\mathrm{Sim}^{C1}(P3, P1) = 0$	$\mathrm{Sim}^{C1}(P4, P1) = 0$	$\mathrm{Sim}^{C1}(P5, P1) = 0$	$\mathrm{Sim}^{C1}(P6, P1) = 0$
$\mathrm{Sim}^{C1}(P1, P4) = 0$	$\mathrm{Sim}^{C1}(P2, P4) = 0$	$\mathrm{Sim}^{C1}(P3, P2) = 0$	$\mathrm{Sim}^{C1}(P4, P2) = 0$	$\mathrm{Sim}^{C1}(P5, P2) = 0$	$\mathrm{Sim}^{C1}(P6, P2) = 0$
$\mathrm{Sim}^{C1}(P1, P4) = 0$	$\mathrm{Sim}^{C1}(P2, P5) = 0$	$\mathrm{Sim}^{C1}(P3, P4) = 0$	$\mathrm{Sim}^{C1}(P4, P3) = 0$	$\mathrm{Sim}^{C1}(P5, P3) = 0$	$\mathrm{Sim}^{C1}(P6, P3) = 0$
$\mathrm{Sim}^{C1}(P1, P4) = 0$	$\mathrm{Sim}^{C1}(P2, P6) = 0$	$\mathrm{Sim}^{C1}(P3, P5) = 0$	$\mathrm{Sim}^{C1}(P4, P5) = 0$	$\mathrm{Sim}^{C1}(P5, P4) = 0$	$\mathrm{Sim}^{C1}(P6, P4) = 0$
---	---	$\mathrm{Sim}^{C1}(P3, P6) = 0$	$\mathrm{Sim}^{C1}(P4, P6) = 0$	$\mathrm{Sim}^{C1}(P5, P6) = 0$	$\mathrm{Sim}^{C1}(P6, P5) = 0$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3s
DOI: https://doi.org/10.17762/ijritcc.v11i3s.6180
***Article Received**: 16 December 2022 **Revised**: 24 January 2023 **Accepted**: 30 January 2023*

Table 1 displays the failure scenarios for CFUB. A similar restriction can be calculated for target customers C2 and C3. The above scenario opens a scope where improvement can be conceivable. However, the proposed work of P.K. Singh et al. has mitigated the limitation of BC to some extent considering the extra parameters. Table 2 shows the customer's interest in the item's categorical attributes, as illustrated by P.K. Singh et al. .

Table 2: Interests of Customers & Behaviour

	Action	Adventure	Animation	Children's	Comedy	Documentary	Crime	Drama	Fantasy	Musical	Romance	Thriller
C1	0.4	0.4	0.2	0	0	0	0	0	0	0	0	0
C2	0	0	0	0.16	0.16	0.16	0.16	0.16	0.16	0	0	0
C3	0	0	0	0	0	0	0	0	0	0.33	0.33	0.33

Here, from tables 1 and 2, we can observe that the proposed work of P.K.Singh et al. cannot determine the similarity between C1-C2, C1-C3, and C2-C3 because ratings and customer's interest in the item's categorical attributes both are disjoint. Motivated by the above discussion, these observations provide a scope to overcome the limitations of existing SMs that solely consider ratings .

3.1. Collaborative Similarity Measurements

Comparison of similarity measurements in recommendation systems is important as it determines the effectiveness and accuracy of the recommendations generated by the system. Some commonly used similarity measurements include: These are just a few of the many similarity measurements used in recommendation systems . The nature of the recommendation system and the information at hand should guide the selection of a suitable similarity metric. Each similarity metric has advantages and disadvantages; selecting the most appropriate one for a certain recommendation system is a matter of considering its needs and objectives.

Cosine similarity measures the degree of similarity between two non-zero vectors in the inner product space. It is calculated by taking the cosine of the angle between the two vectors and is typically represented as a value between 0 and 1 . Cosine similarity has applications in various algorithms and libraries, such as Matlab, SciKit-Learn, and TensorFlow. The benefit is that it can measure similarities between vectors even though their sizes may put them at a large distance apart when calculating the Euclidean distance . An example of cosine similarity would be calculating the similarity between two documents by measuring the cosine of the angle between their respective word vectors. This measure compares the angle between two vectors and determines the similarity between two items based on their features. It is widely used in content-based recommendation systems.

$$\text{Sim}\,(u,v)^{\cos} = \frac{\sum_{i\in I(u,v)} R(u,i)\cdot R(v,i)}{\sqrt{\sum_{i\in I(u,v)} R(u,i)^2}\cdot\sqrt{\sum_{i\in I(u,v)} R(v,i)^2}}$$

Pseudo Code:

```
def cosine_similarity(x, y):
return np.dot(x, y) / (np.linalg.norm(x)
    * np.linalg.norm(y))
```

Here, x and y are two user or item ratings in a vector. Cosine similarity calculates the dot product of two vectors divided by the product of their norms. This gives us a score from -1 to 1 representing the cosine of the angle between the two vectors. If the vectors have a score of 1, they are identical, and if they have a score of -1, they are perpendicular to one another. When the score is zero, the vectors are orthogonal, meaning they are completely dissimilar.

Adjusted cosine similarity is a metric used to measure the similarity between two items in a multi-dimensional space. It is commonly used in recommendation systems to recommend items based on the similarity between users or content. Adjusted cosine similarity returns a number between 0 and 1, where 0 indicates no resemblance and 1 indicates perfect similarity . The adjusted cosine similarity metric considers user rating differences by subtracting each user's average rating from their ratings. This offsets the drawback of using cosine similarity alone, which does not consider differences in ratings of users. In Python, adjusted cosine similarity can be calculated using the Sklearn library. It can also be calculated for more than two items, and the resulting value will decrease if there is a difference in ratings for one item .

$$\text{ACSim}\,(u,v) = \frac{\sum_{i\in I}\left(R_{u,i}-\bar{R}_i\right)\left(R_{v,i}-\bar{R}_i\right)}{\sqrt[2]{\sum_{i\in I}\left(R_{u,i}-\bar{R}_i\right)^2}\sqrt[2]{\sum_{i\in I}\left(R_{v,i}-\bar{R}_i\right)^2}}$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3s
DOI: https://doi.org/10.17762/ijritcc.v11i3s.6180
***Article Received**: 16 December 2022 **Revised**: 24 January 2023 **Accepted**: 30 January 2023*

Pseudo Code:

```
def adjusted_cosine_similarity(x, y, mean_ratings):
    x_mean = np.mean(x)
    y_mean = np.mean(y)
    x_center = x - x_mean
    y_center = y - y_mean
    numerator = np.dot(x_center, y_center)
    denominator = np.sqrt(np.sum(x_center ** 2))
        * np.sqrt(np.sum(y_center ** 2))
    return numerator / denominator
```

Here, x and y are vectors reflecting the ratings of two individuals or objects, respectively, in this code, and mean ratings are the average rating across all users and items. The mean ratings are subtracted from each rating to calculate the cosine similarity using the modified cosine similarity. This is done to counteract the influence of common user tastes and average product evaluations.

Pearson Correlation: This similarity measure calculates the correlation between two items and determines their similarity based on their co-occurrence patterns. It is used in collaborative filtering recommendation systems. The Pearson correlation coefficient (r) is a measure of the strength of a linear association between two variables . It ranges from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation. The Pearson correlation coefficient is used to compare the similarity between two objects. It is calculated by subtracting the mean of each variable from its respective value and then dividing it by the standard deviation of each variable. This normalizes the values of the vectors to their arithmetic mean, which means that the origin of the vector space is considered when calculating similarity.

$$\text{Sim}(u,v)^{\text{COR}} = \frac{\sum_{i \in I(u,v)} (R(u,i) - \overline{R(u)}) \cdot (R(v,i) - \overline{R(v)})}{\sqrt{\sum_{i \in I(u,v)} (R(u,i) - \overline{R(u)})^2} \cdot \sqrt{\sum_{i \in I(u,v)} (R(v,i) - \overline{R(v)})^2}}$$

Pseudo Code:

```
def pearson_correlation(x, y):
    x_mean = np.mean(x)
    y_mean = np.mean(y)
    x_center = x - x_mean
    y_center = y - y_mean
    numerator = np.dot(x_center, y_center)
    denominator = np.sqrt(np.sum(x_center ** 2))
        * np.sqrt(np.sum(y_center ** 2))
    return numerator / denominator
```

Here, x and y are two user or item ratings in a vector. After centring the two variables, x and y, the Pearson Correlation calculates their correlation. For a perfect linear relationship between the two variables, the similarity score would be 1; for a perfect negative linear relationship, it would be -1; and for no linear relationship, it would be 0. This can be used as a metric of similarity in suggestive algorithms.

Euclidean Distance similarity measure calculates the distance between two items and is used to determine their similarity. It is commonly used in content-based recommendation systems. Euclidean distance is a measure of similarity used in data science to compare two lists of numbers (i.e. vectors). It is defined as the square root of the sum of squared differences between corresponding elements of the two vectors . Euclidean distance is most often used to compare profiles of respondents across variables. The Euclidean distance score is calculated by taking the square root of the sum of squared differences between corresponding elements of two vectors. This score can then be converted into a "distance-based similarity" by taking the inverse . This conversion from a distance to a similarity makes sense because it allows for comparison between different vectors without being affected by changes in scale. Euclidean distance works well when dealing with low-dimensional data and when magnitude matters, but other measures have been developed to account for its disadvantages, such as cosine similarity, which is often used to counteract Euclidean distance's problem with high dimensionality .

$$\text{EDSim}(u,v) = \frac{1}{1 + \sqrt[2]{\sum_{i \in I} \left(R_{u,i} - R_{v,i}\right)^2}}$$

Pseudo Code:

```
def euclidean_distance(x, y):
    return np.linalg.norm(x - y)
```

Here, x and y are two user or item ratings in a vector. Using the Euclidean norm, the Euclidean Distance determines how far apart the two vectors are. In other words, we can now quantify how far apart the two vectors are. Distances between people or products are more meaningful in recommendation algorithms when they are closer.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3s
DOI: https://doi.org/10.17762/ijritcc.v11i3s.6180
***Article Received**: 16 December 2022 **Revised**: 24 January 2023 **Accepted**: 30 January 2023*

Remember that the triangle inequality disqualifies the Euclidean distance as a valid similarity metric. It is still useful as a baseline or when the vectors have a Euclidean structure.

Spearman Correlation: Spearman's rank correlation coefficient (ρ, also signified by rs) is a non-parametric measure of rank correlation (statistical dependence between the rankings of two variables). It assesses how well the relationship between two variables can be described using a monotonic function. Spearman's correlation coefficient measures the strength and direction of monotonic association between two variables. The formula for Spearman's rank coefficient is $\boldsymbol{\rho}$ = Spearman's rank correlation coefficient. The Spearman Rank Correlation can take a value from +1 to -1, where +1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation [2] [4]. An example of Spearman's Rank Correlation is given in, where the data shows a weak correlation with a value near 0 .

$$\text{SCSim}\,(u,v) = \frac{\sum_{i\in I}\left(k_{u,i}-\overline{k_u}\right)\left(k_{v,i}-\overline{k_v}\right)}{\sqrt[2]{\sum_{i\in I}\left(k_{u,i}-k_u\right)^2}\sqrt[2]{\sum_{i\in I}\left(k_{v,i}-\bar{k}_v\right)^2}}$$

Pseudo Code:

```
def spearman_correlation(x, y):
return scipy.stats.spearmanr(x, y).correlation
```

Here, x and y are two user or item ratings in a vector. To determine the rank correlation between x and y, the Spearman Correlation is used. A similarity score of 1 indicates a perfect monotonic relationship between the two variables, a similarity score of -1 indicates a perfect inverse monotonic relationship, and a similarity score of 0 indicates no monotonic relationship between the two variables. In recommending systems, this may be used to measure how similar two items are. Also, unlike the Pearson Correlation, which can be affected by outliers and non-linear relationships, the Spearman Correlation is a non-parametric measure.

Manhattan Distance is a distance metric between two points in N-dimensional vector space. It is the sum of the lengths of the projections of the line segment between the points onto the coordinate axes. In simple terms, it is the sum of the absolute difference between the measures in all dimensions of two points. Manhattan distance is also called Taxicab or City Block distance and is related to the L1 vector norm and sum absolute error and mean absolute error metric. It might make sense to calculate the Manhattan distance instead of the Euclidean distance for two vectors in an integer feature space. Manhattan distance has several applications, such as clustering, classification, and pattern recognition. It can also measure the similarity between objects in data mining tasks such as document clustering. Additionally, it can be used to compare strings by calculating how many characters need to be changed for one string to become another string (Hamming Distance) .

.

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

Pseudo Code:

```
def manhattan_distance(x, y):
return np.sum(np.abs(x - y))
```

A pair of users' or objects' ratings are represented by the vectors x and y in this code. The Manhattan Distance takes in two vectors and returns the total of their absolute differences. As a result, we can quantify the degree to which the two vectors differ from one another, with a bigger distance signifying greater dissimilarity. The Manhattan Distance is a distance measure used in recommendation systems; it is also known as the L1 norm or the Taxicab Distance. Since it is more forgiving of data variance and less sensitive to outliers, the Manhattan Distance can be a useful alternative to the more traditional Euclidean Distance.

Jaccard Similarity is a common proximity measurement used to compute the similarity between two objects, such as two text documents. It is calculated by dividing the number of observations in both sets by the number of observations in either set. Jaccard Similarity can be used to find the similarity between two binary variables, two sets, or characters. The value of Jaccard Similarity ranges from 0 to 1, where 0 indicates the documents are identical while 1 means nothing is common among the documents. The mathematical representation of the similarity is as follows: Jaccard Similarity = (number of observations in both sets) / (number in either set) .

$$\text{Sim}\,(u,v)^{\text{Jaccard}} = \frac{(I_u \cap I_v)}{(I_u \cup I_v)}$$

Pseudo Code:

```
def jaccard_similarity(x, y):
intersection = np.sum(np.minimum(x, y))
union = np.sum(np.maximum(x, y))
return intersection/union
```

Both x and y are binary vectors in this code, each indicating whether or not a given item belongs to one of two user sets.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3s
DOI: https://doi.org/10.17762/ijritcc.v11i3s.6180
***Article Received**: 16 December 2022 **Revised**: 24 January 2023 **Accepted**: 30 January 2023*

The Jaccard Similarity measures how similar two sets of data are by determining the proportion of shared data to total data in the union. The result is a numerical representation of the degree to which the two sets are similar, with a value of 1 representing perfect congruence and a value of 0 indicating complete dissimilarity. In information retrieval and recommendation systems, where binary data represents the presence or absence of objects, Jaccard Similarity is frequently utilized.

These commonly used similarity measures have a few limitations and introduce complications when determining the similarity between two users or items. The sparsity problem is a significant constraint on the use of collaborative Filtering. Data sparsity refers to the difficulty in identifying reliable, comparable users, as active users generally only rated a small percentage of items. Thus, even if we have a large amount of data on user ratings, it is useless unless we have items that have been co-rated. When two users are compared, co-rated items are those that both have rated. The requirement for co-rated items arises because most similarity measures use the dot product concept between two vectors. When calculating the similarity between two items or users, if the user has not rated the item, the rating is assumed to be 0; thus, when computing the dot product, the result will be 0. As a result, it requires the existence of two non-zero-rating vectors, colloquially referred to as co-rated vectors. It is reasonable to assume that the number of co-rated items will be quite small compared to the total dataset. Traditional similarity measures are unable to incorporate non-co-rated items. There are a few customized similarity measurement models also considered for comparison.

The main disadvantage of traditional similarity measures is their directionality. For instance, the Euclidean distance determines the distance between the vectors' ends, whereas the cosine similarity determines the similarity of two non-zero vectors in an inner product space. When computing the similarity between users or items in a dataset, these direction-oriented similarity measures encounter a few issues. Let us examine the difficulties that this similarity measures face.

3.2. The outcome of Similarity Measurements

Collaborative Filtering is a popular technique used in recommendation systems to generate personalized recommendations for users. It works by using the similarities between users and items to make recommendations. The basic idea behind collaborative Filtering is to find similar users or items and use their preferences to make recommendations. Collaborative Filtering is a technique that has been widely used in recommendation systems. The traditional similarity measures used in collaborative Filtering to compute the similarity between items or users have several disadvantages, including low performance due to the sparsity of data, cold-start issues, and other issues mentioned previously. To address these issues, researchers have developed new measures of similarity. Most of the proposed similarity measures in these research papers combine several popular ones. Similarity measurement is a key component of collaborative Filtering and is used to determine the similarity between users or items. There are various methods of measuring similarities, such as cosine similarity, Jaccard similarity, Pearson correlation coefficient, Euclidean distance, and Manhattan distance.

Cosine similarity is a widely used measure in content-based recommendation systems. It calculates the cosine of the angle between two vectors, representing the features of two items, and is based on the assumption that items with similar features are likely to be similar. The Jaccard similarity is based on the Jaccard index and is used to determine the similarity between two sets of items. The Pearson correlation coefficient measures the linear relationship between two variables and calculates the correlation between two items based on their co-occurrence patterns. Euclidean distance is a measure of the distance between two items and is used to determine their similarity, while Manhattan distance is a measure of the sum of absolute differences between two items and is used to determine their similarity.

Table 3: Evaluation Matrix of Similarity Measurements

Measure	Equal-Ratio	Unequal Length	Flat-Value	Opposite-Value	Single-Value	Cross-Value	No Co-rated Items exist
Cosine similarity	X	Y	*	X	X	X	X
Adjusted cosine similarity	X	X	*	Y	*	*	X
Euclidean Distance	Y	Y	Y	X	Y	*	X
Pearson Correlation	X	*	X	Y	X	*	X
Spearman Correlation	X	*	X	Y	X	*	X
Manhattan Distance	Y	Y	Y	Y	Y	Y	X
Jaccard Similarity	Y	Y	Y	X	Y	*	X

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 3s
DOI: https://doi.org/10.17762/ijritcc.v11i3s.6180
***Article Received**: 16 December 2022 **Revised**: 24 January 2023 **Accepted**: 30 January 2023*

The evaluation shows that the similarity metrics have their strengths and weaknesses, and the choice between them depends on the specific requirements of the recommendation system. For example, Pearson Correlation Coefficient is good for capturing linear relationships, while Cosine Similarity is good for capturing angular relationships. Jaccard Similarity is good for capturing overlap, while Euclidean Distance is good for capturing distance. The best similarity metric for a particular recommendation system depends on the data type and the problem's nature.

IV. CONCLUSION

Collaborative Filtering (CF) is a type of personalized recommendation method used in Recommender Systems (RS) to filter the flow of data that can be recommended to a target user according to their taste and preferences. There are two main approaches for CF-based RS: User-Based Collaborative Filtering (UBCF) and Item-Based Collaborative Filtering (IBCF). Similarity measures are used in CF-based RS to identify a set of neighbours for the active user on hand. Commonly used similarity measures for UBCF include Cosine, Pearson Correlation Coefficient (PCC), Adjusted Cosine Similarity, and Jaccard Similarity. For IBCF, Cosine Similarity is commonly used. These similarity measures can be implemented on real data sets to evaluate their advantages and disadvantages. In addition, embeddings can be learned automatically in collaborative filtering models. This allows for the dot product of the user embedding and the item embedding to explain the feedback matrix.

In conclusion, collaborative Filtering is a powerful technique for generating personalized recommendations in recommendation systems. Similarity measurement is an important component of collaborative Filtering and is used to determine the similarity between users or items. The choice of similarity measure depends on the specific requirements and goals of the recommendation system, as well as the type of data being used. Choosing the right measure for the task at hand makes it possible to generate accurate and effective recommendations.

REFERENCES

[1] K. Chopra, K. Gupta and A. Lambora, "Future internet: The internet of things-a literature review," IEEE - 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 135-139.

[2] T. R. Mahesh, V. Vivek and V. V. Kumar, "Recommendation Systems: Different Techniques, Challenges and Future Directions," International Journal of Information Technology, Research and Applications, vol. 1, no. 2, pp. 1-8, 2022.

[3] A. Da'u, N. Salim, I. Rabiu and A. Osman, "Recommendation system exploiting aspect-based opinion mining with deep learning method," Information Sciences, vol. 512, pp. 1279-1292, 2020.

[4] P. K. Singh, P. K. Pramanik and P. Choudhury, "Collaborative filtering in recommender systems: Technicalities, challenges, applications, and research trends," Apple Academic Press, pp. 183-215, 2020.

[5] A. A. Amer and L. Nguyen, "Combinations of jaccard with numerical measures for collaborative filtering enhancement: Current work and future proposal," arXiv preprint arXiv:2111.12202, 2021.

[6] V. Mohammadi, A. M. Rahmani and A. M. Darwesh, "Trust-based recommendation systems in Internet of Things: a systematic literature review," Human-centric Computing and Information Sciences, vol. 9, no. 1, pp. 1-61, 2019.

[7] G. Jain, T. Mahara and S. C. Sharma, "Performance Evaluation of Time-based Recommendation System in Collaborative Filtering Technique," Procedia Computer Science, vol. 218, pp. 1834-1844, 2023.

[8] G. R. Lima, C. E. Mello, A. Lyra and G. Zimbrao, "Applying landmarks to enhance memory-based collaborative filtering," Information Sciences, vol. 2020, no. 513, pp. 412-428, 2020.

[9] W. Yue, Z. Wang, B. Tian, M. Pook and X. Liu, "A hybrid model-and memory-based collaborative filtering algorithm for baseline data prediction of Friedreich's ataxia patients," IEEE Transactions on Industrial Informatics, vol. 17, no. 2, pp. 1428-1437, 2020.

[10] D. Valcarce, A. Landin, J. Parapar and Barreiro, "Collaborative filtering embeddings for memory-based recommender systems," Engineering Applications of Artificial Intelligence, vol. 85, pp. 347-356, 2019.

[11] K. Lin, N. Sonboli, B. Mobasher and R. Burke, "Calibration in collaborative filtering recommender systems: a user-centered analysis," Proceedings of the 31st ACM Conference on Hypertext and Social Media, pp. 197-206, 2020.

[12] B.Nagesh, Ch.Harika, " Recommendation System for Find Friend on Social Networks," International Journal of Computer Engineering In Research Trends, vol 2,no 12, pp. 1188-1191 ,2015.

[13] J. P. Yaacoub, H. Noura, O. Salman and A. Chehab, "Security analysis of drones systems: Attacks, limitations, and recommendations," Internet of Things, vol. 11, p. 100218, 2020.

[14] S. A. Olvera and T. B. Núnez, "Design a product recomendation model," 2022.

[15] M. N. Ballesteros Carretero, "Research on recommender systems: A bibliometric study," 2021.

[16] S. Ojagh, M. R. Malek , S. Saeedi and S. Liang, "A location-based orientation-aware recommender system using IoT smart devices and Social Networks," Future Generation Computer Systems, vol. 108, pp. 97-118, 2020.

[17] C. Hansen, L. Maystre and R. Mehrotra, "Contextual and sequential user embeddings for large-scale music recommendation," Proceedings of the 14th ACM Conference on Recommender Systems, 2020.

[18] V. Lytvyn, V. Vysotska, V. Shatskykh and I. Kohut, "Design of a recommendation system based on Collaborative Filtering and machine learning considering personal needs of the user," 2019.

[19] S. Gupta and V. Kant, "Credibility score based multi-criteria recommender system," Knowledge-Based Systems, vol. 196, p. 105756, 2020.

[20] D. Algawiaz, "Personalized Risk Aware Recommender Systems," Doctoral dissertation, ResearchSpace@ Auckland, 2020.

[21] S. R. Patil ," Hybridization Of Web Page Recommender Systems Based On Ml Techniques," International Journal of Computer Engineering In Research Trends, vol 6,no 5, pp. 310-316 ,2019.

[22] R. Logesh, V. Subramaniyaswamy and D. Malathi, "Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method," Neural Computing and Applications, vol. 32, pp. 2141-2164, 2020.

[23] M. F. Aljunid and D. H. Manjaiah, "Movie recommender system based on collaborative filtering using apache spark," Springer Singapore - Data Management, Analytics and Innovation: Proceedings of ICDMAI 2018, vol. 2, pp. 283-295, 2019.

[24] K. Han, "Personalized news recommendation and simulation based on improved collaborative filtering algorithm," Complexity, pp. 1-12, 2020.

[25] P. F. Cueto, M. Roet and A. Słowik, "Completing partial recipes using item-based collaborative filtering to recommend ingredients," arXiv preprint arXiv:1907.12380, 2019.

[26] A.Kirankumar,P.Ganesh Kumar Reddy, A.Ram Charan Reddy, Baneti Shivaji and D.Jayanarayan Reddy," A Logic-based Friend Reference Semantic System for an online Social Networks," International Journal of Computer Engineering In Research Trends, vol 1,no 6, pp. 501-506 ,2014.

[27] B. Alhijawi and Y. Kilani, "A collaborative filtering recommender system using genetic algorithm," Information Processing & Management, vol. 57, no. 6, p. 102310, 2020.

[28] H. Kusniyati and A. A. Nugraha, "Analysis of Matric Product Matching Between Cosine Similarity with Term Frequency-Inverse Document Frequency (TF-IDF) and Word2Vec in PT," Pricebook Digital Indonesia, Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technology, 2020.

[29] J. Mlyahilu and J. N. Kim, "Parametric and Non Parametric Measures for Text Similarity," Journal of the Institute of Convergence Signal Processing, vol. 20, no. 4, pp. 193-198, 2019.

[30] M. Kirişci, "New cosine similarity and distance measures for Fermatean fuzzy sets and TOPSIS approach," Knowledge and Information Systems, pp. 1-14, 2022.

[31] N.Satish Kumar, Sujan Babu Vadde," Typicality Based Content-Boosted Collaborative Filtering Recommendation Framework," International Journal of Computer Engineering In Research Trends, vol 2,no 11, pp. 809-813 ,2015.

[32] X. Wang, Y. Ran and T. Jia, "Measuring similarity in co-occurrence data using ego-networks," Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 30, no. 1, p. 013101, 2020.

[33] Yedukondalu, Gangolu & K., Samunnisa & Bhavsingh, M. & Raghuram, I & Addepalli, Lavanya," MOCF: A Multi-Objective Clustering Framework using an Improved Particle Swarm Optimization Algorithm",International Journal on Recent and Innovation Trends in Computing and Communication,Vol 10,no 10. Pp.143-154, 2022. 10.17762/ijritcc.v10i10.5743.

[34] J. Wang and Y. Dong, "Measurement of text similarity: a survey," Information, vol. 11, no. 9, p. 421, 2020.

[35] S. A. Shishavan, F. K. Gündoğdu and K. Farrokhizade, "Novel similarity measures in spherical fuzzy environment and their applications," Engineering Applications of Artificial Intelligence, vol. 94, p. 103837, 2020.

[36] Y. Xu and P. Wang, "An enhanced squared exponential kernel with Manhattan similarity measure for high dimensional Gaussian process models.," nternational Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 85390 , p. V03BT03A025, 2021.

[37] M. T. Hanif, M. Yongtong and S. B. Shah, "Economics of open-access fisheries: A case of factors affecting the revenue of coastal or inshore longline fisheries in Pakistan," 2021.

[38] A. Golovanov, A. Kupavskii and A. Sagdeev, "Odd-distance and right-equidistant sets in the maximum and Manhattan metrics," European Journal of Combinatorics, vol. 107, p. 103603, 2023.

[39] Joy and V. G. Renumol, "Comparison of generic similarity measures in E-learning content recommender system in cold-start condition," 2020 IEEE Bombay section signature conference, pp. 175-179, 2020.