International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 04 | Apr 2023 www.irjet.net p-ISSN: 2395-0072

Analysing the performance of Recommendation System using different similarity metrics

Kopanathi Sonali[1], S. V. G. Reddy[2], K. Thammi Reddy[3], V. Valli Kumari[4]

[1]M. Tech Student, Department of Computer Science and Engineering, GITAM University, Visakhapatnam, Andhra Pradesh, 530045, India.
[2]Associate Professor, Department of Computer Science and Engineering, GITAM University, Visakhapatnam, Andhra Pradesh, 530045, India.
[3]Professor, Department of Computer Science and Engineering, GITAM University, Visakhapatnam, Andhra Pradesh, 530045, India.
[4]Professor, Department of CS and SE, College of Engineering (Andhra University), Visakhapatnam, Andhra Pradesh, 530045, India

---

Abstract - In today's modern era of information technology, finding a favourite item in a large dataset has become an essential issue. So, there is a need for a more effective recommendation system with better performance. To achieve this, a Collaborative filtering recommendation system is proposed in this work. Here, the comparison is made with various similarity metrics like Pearson Correlation, Cosine Similarity, Jaccard Coefficient, MSD (Mean Squared Difference), Sorensen Dice Coefficient and SVD (Singular Value Decomposition) on the Movielens 100k dataset. It is observed that the Jaccard Similarity metric, compared to Pearson correlation and cosine similarity, produces better outcomes with improved accuracy and less time complexity.

***Key Words*: Recommended System, Jaccard Index, Pearson Correlation, Cosine Similarity, Spearman rank Correlation, Sorensen Dice coefficient, Prediction, Recommendations.**

1. INTRODUCTION

A type of system known as a recommendation system is utilized for filtering or sorting information with the purpose of predicting a user's preference or rating for a specific item. These systems are commonly used to provide suggestions for items such as books, TV shows, movies, music, and apps that may be of interest to a group of users.

To generate recommendations, the system analyses users' past interests, which can be gathered either explicitly, through user ratings of items, or implicitly, by tracking user behaviour like purchasing history, browsing data, and downloaded applications. In addition, the system may utilize information from the user's profile, such as age, gender, nationality, preferences, and habits of their group of users, to compare and present personalized recommendations.

Fig.1. Recommendation System

In the depicted scenario, a user's purchase of a camera triggers a recommendation from the system to purchase a tripod. The recommendation uses the user's previous actions and preferences as a basis, which are used to suggest products that align with their interests. This is an example of how recommendation systems operate.

1.1 Types

In terms of recommendation techniques, there exist various types including:

Content-based:

This type of recommendation system relies on user reviews, ratings, and product features to generate recommendations. To find similar items to those previously liked by the user, the system calculates the similarity between items based on their associated features. The system then recommends items to users who have shown similar preferences. Recommendations are generated by evaluating similarities among items and considering the top-most regular items amongst neighbours.

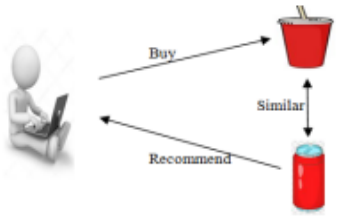


Fig.2. Content-Based Filtering

In the given illustration, a user's purchase of juice prompts the recommendation system to suggest purchasing coke based on the similarity of the items. Recommendations are made by taking into account the user's previous behaviour and preferences, demonstrating how content-based recommendation systems function.

Collaborative filtering:

Collaborative filtering is a strategy used to suggest items to users by finding those with similar preferences and recommending items they have previously preferred. The system evaluates the similarity of users' preferences by examining their rating history, also referred to as "people-to-people correlation." This approach is commonly employed in Recommender Systems and is implemented using various methods.

Neighbourhood methods and item-item approaches are two strategies that focus on the relationships between items or users in Recommender Systems. The item-item approach models a user's preference for an item based on their ratings of similar items. However, the nearest-neighbours method is more widely used due to its efficiency, simplicity, and capacity to generate precise and personalized recommendations, particularly for smaller datasets. Additionally, a variety of collaborative filtering algorithms are available to accommodate larger datasets with numerous users and a greater number of products than items.

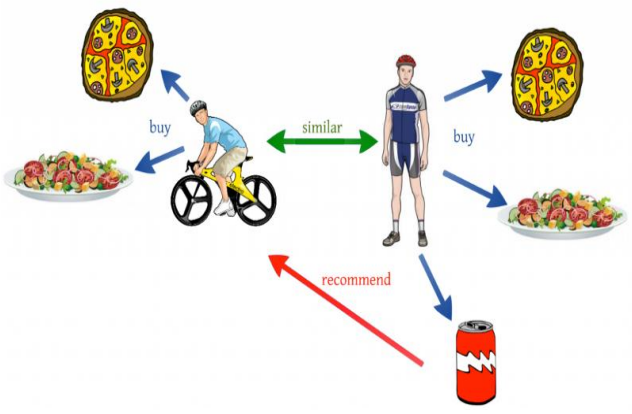


Fig.3. Collaborative Filtering

As shown in the above figure, User A orders Salad and pizza, while User B likes to order Salad, pizza and coke. Both users have ordered several times from the same food ordering app and are given high ratings for their preferred items. The collaborative filtering system identified that User B has similar preferences to User A and suggested some of their favourite items to User A, who may be more likely to enjoy those items based on their past behaviour. Collaborative filtering helps to personalise the recommendations and make them more relevant and appealing to each User. Hence, this is how the collaborative filtering recommended system works.

Hybrid recommender systems:

Recommender systems that combine content-based and collaborative filtering techniques are referred to as hybrid systems. This approach aims to take advantage of the strengths of each technique while addressing its limitations. The hybrid system works by using the strengths of one technique to overcome the weaknesses of the other.

For instance, collaborative filtering systems face challenges when recommending new items that users still need to rate. However, the content-based approach does not face this limitation since it relies on item features and descriptions, which are typically readily available. By combining the two techniques, the hybrid recommender system can overcome the limitations of each technique and provide more accurate and personalised recommendations to users.

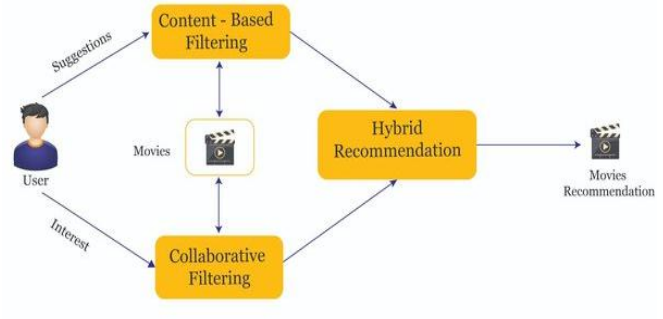


Fig.4. Hybrid Recommender system

As depicted in the preceding image, hybrid recommender systems utilize the advantages of content-based and collaborative filtering methods to offer users more precise and varied recommendations.

1.3 Collaborative Filtering approach

User-to-User Collaborative filtering approach.

The user-to-user Collaborative Filtering approach is widely used for generating recommendations based on the preferences of other users who share similar interests with

the target user. It assumes that users who have rated similar items in the past will likely rate future items similarly as well.

To generate recommendations, the system first identifies the most similar users to the target user based on their past ratings. Then, it considers the ratings of those similar users for items that the target user has not yet rated to predict their potential ratings. Finally, the system generates a list of top recommendations for the target user based on those predicted ratings.

In neighbourhood algorithms, the system selects a subset of users similar to the target user based on a similarity metric such as cosine similarity. The system then computes a weighted average of the ratings of those selected users to generate predictions for the target user. The weights assigned to each user are typically based on their similarity to the target user.

Item-to-Item Collaborative filtering approach

The item-to-item Collaborative Filtering approach recommends items to users based on the ratings that users have given to specific items. Instead of focusing on the preferences of other users, this approach analyses the similarity between the target item and the collection of items that the user has already rated.

To generate recommendations, the algorithm uses similarity measures to identify the k most similar items to the target item. It then computes the similarities and commonalities between the selected items to generate recommendations for the target user based on their past ratings. This approach differs from the user-to-user Collaborative Filtering algorithm, which focuses on identifying users with similar preferences to the target user.

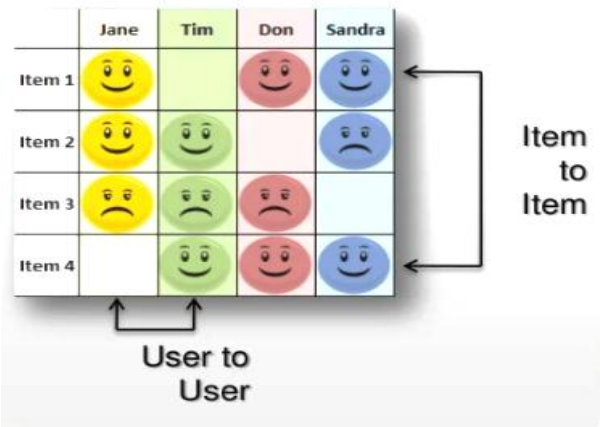


Fig.5. Difference between User-to-User and Item-to-Item

As shown in Figure 5, there are two main approaches for generating recommendations: User-to-User and Item-to-Item Collaborative Filtering.

In the User-to-User approach, recommendations are made by identifying users who have similar preferences to the target user. For example, if Jane and Tim both liked Item 2 and disliked Item 3, it suggests that they may have similar tastes. Therefore, Item 1 might be a good recommendation for Tim. However, this approach may not be scalable for millions of users.

On the other hand, the Item-to-Item approach recommends items based on their similarity to other items that users have liked. For instance, if Tom and Sandra both liked Item 1 and Item 4, it suggests that people who liked Item 4 will also like Item 1, and Item 1 will be recommended to Tim. This approach is scalable to millions of users and items, making it a preferred choice for large-scale recommendation systems.

Hence, the user-to-user approach can be practical for smaller datasets but can become computationally expensive when dealing with large amounts of data. In contrast, the item-to-item approach is more scalable and can handle larger datasets by focusing on item similarity.

2. LITERATURE REVIEW

Resnick et al. [1] presented the user-to-user approach to collaborative filtering, where recommendations for an active user are generated by finding users with similar historical rating behaviours and using their ratings of items that, the active User has not yet seen. The similarity between users can be calculated using various metrics like cosine similarity or Pearson correlation coefficient, which are based on the users' historical rating behaviours. After identifying similar users, the weighted average of their ratings can be calculated, with the weights determined by how similar each User is to the active User.

Breese et al. [2] proposed a prediction problem in collaborative filtering and conducted an empirical analysis of prediction algorithms. The algorithm aims to predict the rating an active user will give an active item. To generate recommendations, the algorithm relies on historical data of ratings and associated content of both users and items. This data is used to make predictions about potential user preferences and item popularity. The active User is the User for whom the prediction is made, and the active item is predicted. By analysing the logged data of user-item interactions, these algorithms estimate the rating the active User would give the active item.

In a study by Herlocker et al. [3], the authors explored the critical decisions involved in evaluating collaborative filtering recommender systems. These included the selection of user assignments to be evaluated, the types of analysis

and datasets used, and the methods of measuring prediction quality. To analyse the performance of different accuracy metrics on a single domain, the authors reviewed previous research and conducted their own experiments. They classified the different accuracy metrics into three equivalence classes and found that the metrics within each class were highly correlated. However, they also found that the metrics across different classes were not correlated with each other. The authors emphasized the importance of considering multiple evaluation techniques to provide a comprehensive understanding of the system's performance.

In a study by Deshpande and Karypis et al. [4] proposed a method for the model-based recommendation that involves evaluating similarities between the items based on their attributes or features. They first identify how to calculate similarity between pairs of items using various methods, such as cosine similarity or Pearson correlation coefficient. Subsequently, they aggregate these similarities to compute an overall similarity score between a recommended candidate item and an item the user has already purchased. By using this method, the system generates personalised recommendations for active users based on their past interactions.

Sarwar [5] conducted an analysis of different algorithms used for generating item-based recommendations. This analysis included examining techniques for computing item-item similarities and strategies for obtaining accurate predictions. To evaluate the effectiveness of these algorithms, they were compared to the basic k-nearest neighbour approach. Based on the results of the study, item-based algorithms were found to have better performance in terms of time compared to user-based CF algorithms. However, the study also found that user-based CF algorithms provided better quality recommendations. The study highlights the importance of considering both performance and recommendation quality when selecting an algorithm for generating recommendations.

Fkih et al. [6] review and compare similarity measures used in Collaborative Filtering-based Recommender Systems. It categorises the most common similarity measures and provides an overview. The authors perform experiments to compare these measures using popular datasets and evaluation metrics. They find that the optimal similarity measure depends on the dataset and evaluation metric. The paper also recommends the most suitable similarity measure for each dataset and evaluation metric. This study is a valuable resource for researchers and practitioners in recommender systems who want to select the most appropriate similarity measures for their CF-based systems.

Bell and Koren et al. [9] proposed a scalable approach for collaborative filtering using jointly derived neighbourhood interpolation weights. Their approach addressed the limitations of traditional collaborative filtering algorithms, such as high computational costs and data sparsity, by deriving neighbourhood weights using a combination of user-to-user and item-to-item approaches. The proposed method was more accurate and scalable than traditional approaches on benchmark datasets. The paper contributes to developing collaborative filtering algorithms for large-scale recommendation systems.

In their study, Saranya et al. [10] compare the performance of various similarity measures for the Collaborative Filtering (CF) technique. Four similarity measures, including Pearson Correlation Coefficient, Cosine Similarity, Mean Squared Difference, and Adjusted Cosine Similarity, were evaluated using the MovieLens dataset. The study aimed to determine which similarity measure provides the best performance for the CF technique. The evaluation is based on three metrics: MAE, RMSE and Precision. The result shows that the Pearson Correlation Coefficient and the Cosine Similarity perform better than the other measures regarding accuracy and Precision. The study provides insights into selecting appropriate similarity measures for CF-based recommendation systems.

3. Problem Identification & Objectives

3.1 Problem statement

The primary objective is to determine the most efficient method of computing similarity between users and items in a dataset. Various similarity measures, including the Jaccard coefficient, cosine similarity and correlation-based similarity, are used to calculate similarity and generate top-N recommendation lists. The goal is to identify which similarity measure offers the quickest and most efficient output for recommending items.

3.2 Motivation:

The increasing number of users on online sites such as Amazon and Netflix have led to a large user-item matrix, requiring the efficient and quick generation of personalised recommendations. However, traditional collaborative filtering systems need help producing high-quality recommendations for users in the shortest time possible, especially for large datasets. Therefore, this study aims to compare the efficiency and advantages of different similarity measures, including the Jaccard coefficient, cosine similarity, Pearson correlation, MSD and adjusted cosine similarity, in generating a top-N recommendation list.

This study aims to identify the most efficient similarity measure that can generate accurate recommendations even for large datasets. User-based top-N recommendation algorithm is also explored as a potential solution to the challenges faced by traditional collaborative filtering

systems. Ultimately, the recommended items to users should satisfy their preferences and interests.

3.3 Objectives

The main goals of this study are:

1) To enhance the accuracy of the proposed recommendation system.

2) To reduce the time complexity of the system.

3) To assess and contrast the performance of various similarity metrics in a recommendation system.

4) To identify the strengths and weaknesses of different similarity metrics in generating recommendations.

4. PROPOSED SYSTEM METHODOLOGY

The proposed system methodology uses a collaborative filtering approach. The system will collect and analyse data from the user's profile, location, and interests. It will then determine parameters that can be used to compare this data with other members' data in the database. The system will search for similarities between users and other members, such as shared interests or locations. The steps to achieve the study's objectives:

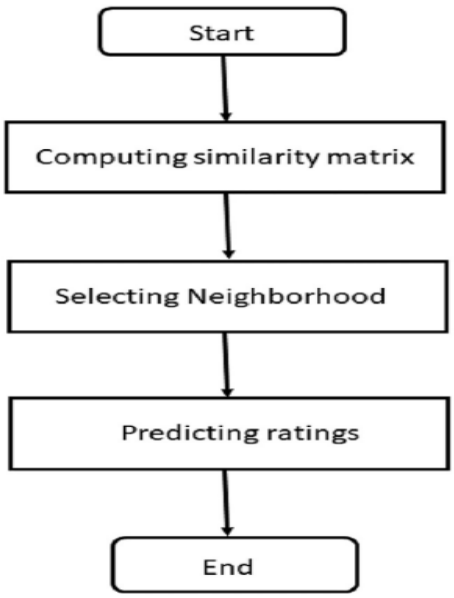


Fig.6. Flowchart of the Collaborative Filtering Approach

In Fig.6, it describes the flow of collaborative filtering approach and its major functionality.

Process:

To build a movie recommender system using user-to-user collaborative filtering method with the help of different similarity metrics.

1) **Prepare the data:** Merge the movie and rating data into a single data frame, pivot the table to create a matrix of users and their movie ratings, and fill any missing values with zeros.

2) **Compute user similarities:** Calculate the similarity between each pair of users based on their movie ratings.

3) **Find similar users:** identifying users who share similar preferences to the target user by selecting the k users with the highest similarity scores.

4) **Predict ratings:** To generate a prediction for each movie that a user has not yet rated, the algorithm calculates a weighted average of the ratings given to that movie by the k most similar users.

5) **Generate recommendations:** Recommend the top n movies with the highest predicted ratings to each user.

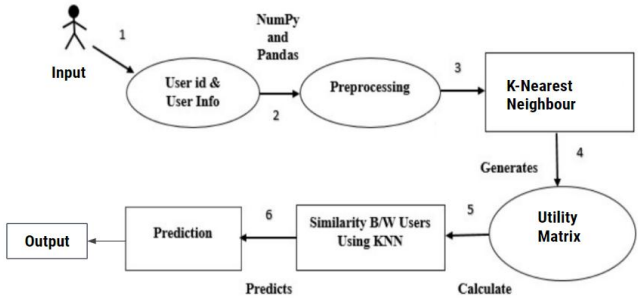


Fig.7. Process Flow Diagram

The process flow diagram describes the steps involved in a recommendation system that uses the K Nearest Neighbour (KNN) approach to find similar users and recommend items to them. The steps involved are:

1) **User Input:** The user provides their User ID and some Facts such as gender, age, and pin code.

2) **Data Pre-processing:** The raw data is pre-processed using NumPy and Pandas libraries. This step involves cleaning and transforming the data into separate frames that can be used for further analysis.

3) **K Nearest Neighbour (KNN):** The KNN approach finds similar users within a community/group. It involves selecting a value for k (the number of nearest neighbours to consider) and computing the distance between users.

4) **Utility Matrix:** A utility matrix is created after applying the KNN approach. This matrix defines the average rating the user gives each other.

5) **User Similarity:** User similarity is calculated using the utility matrix and Pearson correlation. This step involves calculating the similarity between users to determine how similar their preferences are.

6) **Recommendation:** The system uses the utility matrix and suggests items to the user based on the ratings of user's similarity. The recommended items have received high ratings from similar users but have yet to be interacted with by the user.

Hence, the process flow diagram describes a recommendation system that uses collaborative filtering to recommend items to users based on their similarity to other users within a community/group.

Properties of similarity metrics

1) Non-negativity: The similarity between any two objects must be non-negative.

S (X, Y) ≥ 0 for all X and Y.

2) Symmetry: The similarity between two objects should be symmetric.

S (X, Y) = S (Y, X) for all X and Y.

3) Reflexivity: The similarity between an object and itself should be maximum.

S (X, X) = 1 for all X.

4) Triangle inequality: The similarity between two objects x and z should be less than or equal to the sum of their similarities with a third object Y.

S (X, Z) ≤ S (X, Y) + S (Y, Z) for all X, Y, and Z.

5) Range normalisation: Similarity measures should be normalised to a certain range, often [0,1], to be compared across different datasets or applications.

Similarity Metrics

Pearson Correlation:

Pearson Correlation Coefficient (PCC) is used to measure the linear relationship between the two variables. In the context of recommendation systems, these two variables are the ratings two users give to the same items. The PCC value ranges between (-1, +1). Where,

-1 = perfectly negative correlation.

0 = no correlation.

+1 = perfect positive correlation.

A positive value represents a positive correlation or a relationship in which two variables move in the same direction. It is used when

(1) Linear relationship,

(2) Both variables are quantitative,

(3) Normally distributed

(4) They Have no outliers.

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

Advantages of Pearson correlation coefficient

1) It is an accurate method of computing the correlation between two variables.

2) The coefficient helps to determine the degree and strength of the correlation between the variables.

3) It is standardised, allowing for direct comparison between different datasets and variables.

4) It is robust to small amounts of noise or outliers in the data.

5) It can identify both positive and negative correlations between variables.

Disadvantages of Pearson correlation coefficient

1) Pearson's correlation coefficient (PCC) is unsuitable for testing attributive research hypotheses involving only one variable. It is because PCC is a bivariate statistical model that analyses the relationship between two variables.

2) It cannot determine the nonlinear relationships between variables.

3) It does not distinguish between dependent and independent variables.

4) PCC is sensitive to outliers in the data, which can significantly affect the calculated correlation value.

Cosine similarity:

Cosine similarity is a popular similarity measure utilized in Collaborative Filtering to evaluate the similarity between

two sets of rating vectors, which could be the rating vectors of two users or two items. It calculates the cosine of the angle between the two vectors in a multi-dimensional space. By comparing the cosine similarity between two rating vectors, the Collaborative Filtering approach can identify users who share similar tastes or items that exhibit similar rating patterns. The cosine similarity score is bounded between -1 and 1, where a score of 1 represents identical vectors, -1 signifies diametrically opposed vectors, and 0 indicates orthogonal or independent vectors.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Advantages of cosine similarity:

1) Cosine similarity is computationally efficient and does not require a lot of memory, which makes it suitable for large datasets.

2) It is scale-invariant, which means that it is not affected by the magnitude of the ratings, only their directions. This property makes it useful for handling sparse data and data with missing values.

3) It is widely used in many applications, including text mining, image analysis, and recommendation systems.

Disadvantages of cosine similarity:

1) It does not take into account the magnitude of the ratings, only their directions.

2) This can lead to inaccuracies if some users tend to rate items much higher or lower than others.

3) It assumes that the ratings are distributed uniformly across all dimensions, which may not be the case in some datasets.

4) It is sensitive to outliers, which can have a significant impact on the similarity scores if they are not handled properly.

Adjusted cosine similarity:

It is a modification of the cosine similarity measure that addresses its limitations. In Collaborative filtering, similarity between users is computed based on the rating matrix. In contrast, in item-to-item Collaborative filtering, the similarity is computed based on the columns. However, the standard cosine similarity measure used in item-to-item Collaborative filtering does not consider user rating behaviour differences.

$$sim\left(\vec{a}, \vec{b}\right) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2}\sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}}$$

Advantages of Adjusted-cosine similarity

1) Overcomes the drawback of cosine-based similarity

2) It subtracts the user average from each co-rated pair.

3) It considers the differences in rating scales across users.

4) It effectively handles sparse data, where many entries in the user-item matrix are missing.

Spearman Rank Correlation

Spearman Rank Correlation is a similarity measure that computes similarity based on rankings rather than ratings, thus eliminating the need for rating normalization. However, this method is not suitable for incomplete orderings, even if the ratings are comparable.

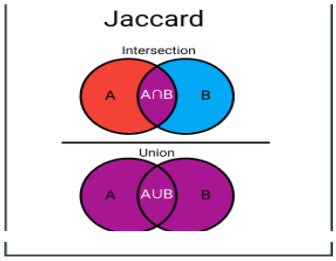
$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)}$$

Mean Squared Difference

The Mean Squared Difference (MSD) approach considers the absolute ratings by calculating the mean of the squared differences between ratings instead of the traditional approach of considering the total standard deviations. This mean is used to determine the similarity between two vectors. A smaller value of the mean squared difference indicates a higher similarity between the vectors.

Jaccard Similarity Coefficient:

The Jaccard coefficient is a metric used to quantify the similarity and dissimilarity between two sample sets. It is computed by dividing the size of the intersection of the sets by the size of their union. When both sets are empty, the coefficient is 1. The coefficient ranges from 0 to 1, where 0 implies that there is no overlap between the sets, and 1 indicates that the sets are identical.



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Where, J = Jaccard Similarity

A = Set 1

B = Set 2

Advantages of the Jaccard Index:

1) Measures the similarity between two asymmetric binary vectors or sets.

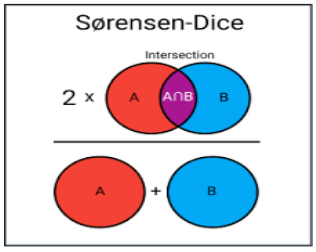
2) This similarity measure is beneficial when duplicates are not necessary.

Disadvantages of the Jaccard Index:

The Jaccard index has a significant disadvantage, mainly when applied to large datasets, as the data size strongly affects the index. In such cases, a slight change in the union can significantly impact the index while keeping the intersection the same.

Sorensen–Dice coefficient:

The Sorensen-Dice coefficient is a similarity measure used to compare sample sets. It is also referred to as the Sorensen-Dice index, Dice's coefficient, or Dice similarity coefficient. This coefficient is closely related to the Jaccard index and is computed by dividing twice the intersection of two sets by the sum of their sizes.



The Sorensen-Dice coefficient computes a value between 0 and 1, where 0 represents no overlap between two sets and 1 indicates complete overlap. Unlike the Jaccard index, the Sorensen-Dice index is more easily understood as the percentage of overlap between the two sets. This index is typically used to measure the similarity of two samples, particularly in the case of discrete data.

$$DSC = \frac{2|A \cap B|}{|A| + |B|}$$

Where

DSC = Dice Similarity Coefficient

|A|, |B| = The num of elements in each set.

|A ∩ B| = The Common num of elements

Disadvantages of Sorensen–Dice coefficient:

1) It weights each item differently based on the size of the relevant set instead of treating them equally.

2) It does not satisfy the triangle inequality. It is considered as a semi-metric version of the Jaccard Index.

4.3 Difficulty of the User-to-User Collaborative Filtering Algorithms

User-to-User collaborative filtering algorithms have become popular in many domains but have several things that could be improved that make them more challenging to use effectively. These include sparsity in the data, scalability issues, lack of diversity in recommendations, the cold start problem, and privacy concerns. While these algorithms have been successful, there is still a need to address these challenges to improve their accuracy and effectiveness.

4.4 Proposed Improvement

An approach has been proposed to improve the challenges faced by user-to-user collaborative filtering algorithms. It analyses the user-item matrix using different similarity metrics to identify similarities and relationships among different products. The Jaccard index similarity metric provides more accurate recommendations to users in less time.

Similarity measures like Pearson's correlation coefficient and cosine similarity are limited in their ability to make recommendations and are susceptible to the cold-start problem. The proposed approach uses co-related and non-related ratings to improve performance and enhance the quality of recommendations.

5. Implementation

This segment explains implementing a movie recommendation system using Python programming with K-

Nearest Neighbour. Implementing the system involves several sub-sections:

Dataset: The MovieLens 100k dataset is a popular dataset used as a standard benchmark in the field of recommender systems. It comprises 100,000 ratings of 1,682 movies, provided by 943 users. The ratings are scored on a scale of 1 to 5, with 5 being the highest. Demographic data about the users, including their age, gender, occupation, and zip code, is also included in the dataset, as well as details about the movies, such as the movie title, release year, and genre.

Data Cleaning: Before building the model, the data must be pre-processed to clean and transform it into a usable format. It involves handling missing data, removing duplicates, and transforming it into a matrix form suitable for analysis.

Model Analysis: Once the data has been cleaned and pre-processed, it is necessary to analyse it to gain insights and determine the appropriate model. It involves exploring the relationships between variables, identifying trends, and selecting relevant features.

Model Building: In this case, the K-Nearest Neighbour algorithm predicts movie ratings based on similar users' preferences. The model is trained on the MovieLens 100k dataset, and its performance is evaluated using various metrics such as RMSE and MAE.

Finally, the model is used to make predict ratings for non-rated items and the results are displayed to the user in the form of recommended movies.

5.1 Technology

Python: It is a dynamically typed programming language used for web development, software, automation, data analysis, and visualisation. It is easy to learn and used by non-programmers for everyday tasks.

NumPy: NumPy is a Python library that enables the creation and manipulation of multi-dimensional arrays and matrices, with a rich set of high-level mathematical functions to facilitate tasks such as linear algebra, Fourier transforms, and array manipulation.

Pandas: A Python library for working with data sets, including analysis, cleaning, exploration, and manipulation. It provides fast and expressive data structures.

Matplotlib: Matplotlib is a Python library that facilitates the creation of visualizations in various forms, including static, animated, and interactive graphics.

Scikit-learn: A free machine-learning library with tools for statistical modelling and machine learning.

Surprise: A Python library for building and evaluating recommender systems. It supports collaborative filtering and matrix factorisation techniques, parallel processing, and hyperparameter tuning.

Google Collaboratory: Google Collaboratory is a cloud-based Python Integrated Development Environment (IDE) launched in 2017 by Google, which provides data scientists with a platform to develop machine learning and deep learning models. It offers cloud storage capabilities to store and share notebooks, datasets, and other files with collaborators.

6. Results and Discussions

6.1 Discussion

Example 1: Sample Data (5*5)

The sample dataset consists of ratings for five items given by five users where Item 5 is not rated by user Alice.

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	5	3	4	4	**?**
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

Tab.1 User rating for 5 Items

Tab.1 represents the user ratings for five items, where Alice did not provide a rating for Item 5. This study aims to predict the rating for the non-rated item using the "User-Based Collaborative Filtering Approach". It uses different similarity metrics, such as Pearson Correlation Coefficient, Jaccard Similarity Coefficient, and Sorensen Dice Coefficient, to determine user similarity. Based on the similarity score, the non-rated item rating is predicted.

Similarity comparison

Users	Pearson Correlation Coefficient	Jaccard Similarity Coefficient	Sorensen Dice Coefficient
Sim (Alice, User1)	0.85	0.2	0.25
Sim (Alice, User2)	0.70	0.6	0.5

Sim (Alice, User3)	0.00	0.5	0.5
Sim (Alice, User4)	- 0.79	0.2	0.25

Tab.2 Users Similarity Comparison by using different similarity metrics

Tab.2 compares the similarity scores between Alice and the other users in the data using three different similarity metrics: Pearson Correlation Coefficient, Jaccard Similarity Coefficient, and Sorensen Dice Coefficient. The table shows that the similarity scores between Alice and User1 and User2 are relatively high, indicating a stronger similarity. However, the Jaccard Similarity Coefficient and Sorensen Dice Coefficient give higher scores for Alice's similarity with User2 and User3. Based on these similarity scores, the KNN algorithm will predict the rating for item 5.

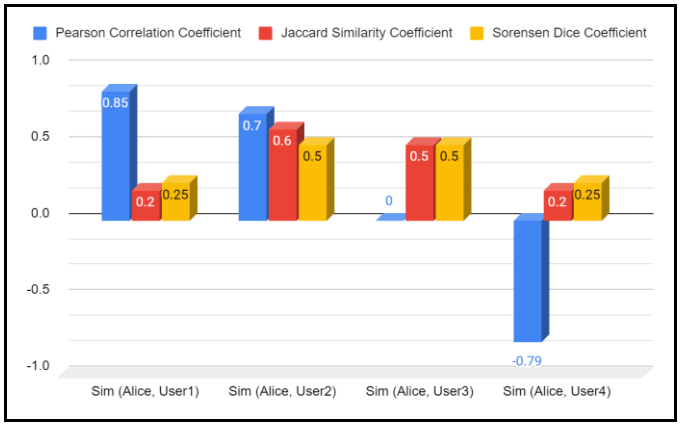


Fig.8 Shows the bar chart comparing user similarity scores using different similarity metrics.

In fig.8, the x-axis shows the users, while the y-axis shows 3 the similarity score. Three different similarity metrics are used to calculate the similarity between Alice and other users. The bar chart shows that user 1 has the highest similarity score using the Pearson Correlation coefficient. Whereas, User 2 has the highest similarity score using Jaccard Similarity Coefficient. User 3 and user 4 have low similarity scores with Alice using Pearson Correlation Coefficient but have average similarity scores using Jaccard Similarity Coefficient and Sorensen Dice Coefficient.

A common prediction function:

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)}$$

	Pearson Correlation Coefficient	Jaccard Similarity Coefficient	Sorensen Dice Coefficient
Pred(Alice, Item5)	2.25	**3.91**	3.67

Tab.3 Comparison of Prediction Value using different similarity metrics.

Tab. 3 shows that the predicted rating value for Alice and Item 5 is highest using the Jaccard Similarity Coefficient, followed by Sorensen Dice Coefficient, and then Pearson Correlation Coefficient.

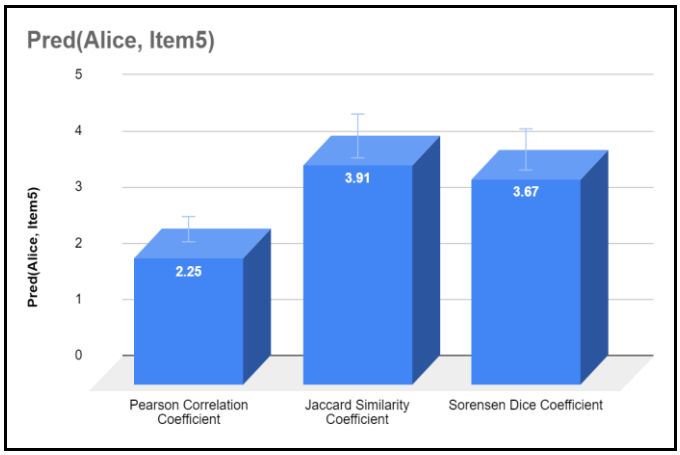


Fig.9 Shows the bar chart comparing the predicted value for non-rated items.

In Fig.9, a comparison of the predicted rating values for the non-rated item (Alice, Item 5) is presented using three different similarity metrics. The Jaccard Similarity Coefficient is observed to yield the highest predicted rating value for (Alice, Item 5), with a value of 3.91.

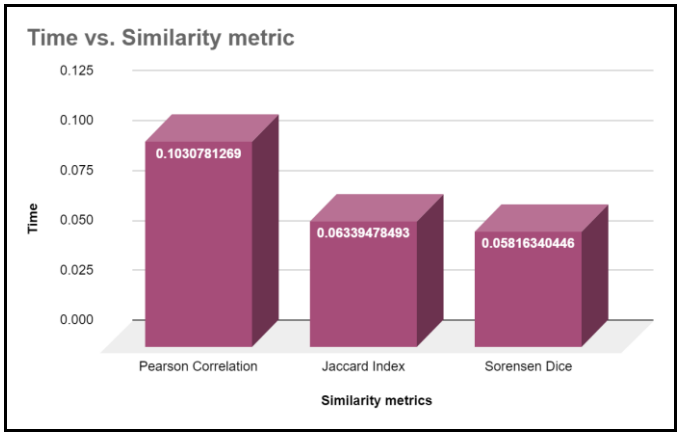


Fig.10 Shows the comparison of execution time for different similarity metrics.

Fig.10 displays a bar chart where the x-axis shows the similarity metrics, and the y-axis shows time. The chart indicates that the Jaccard Similarity Coefficient takes less time than the Pearson Correlation Coefficient.

Example 2: Movielens 100k Dataset

The MovieLens 100K dataset is widely used as a standard reference for evaluating collaborative filtering recommender systems. It comprises of ratings given by users to movies on a 5-point scale, ranging from 1 to 5. The dataset contains 100,000 ratings and is made up of 943 unique users and 1682 unique movies. Additionally, each user in the dataset has rated a minimum of 20 movies.

When evaluating a filtering technique, statistical accuracy metrics are employed to measure accuracy. These metrics directly compare the predicted ratings with the actual user rating. Commonly used statistical accuracy metrics include Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Correlation.

MAE is the most popular and widely used. It measures the deviation of the recommendation from the user's specific value.

$$\mathrm{MAE} = \frac{1}{N}\sum_{u,i}|p_{u,i} - r_{u,i}|$$

Root Mean Square Error (RMSE) metric gives more weight to larger deviations in prediction and is used to evaluate the accuracy of recommendation engines. The lower the RMSE value, the higher the accuracy of the prediction.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{u,i}(p_{u,i} - r_{u,i})^2}$$

6.2 Results :

Comparision Table

Similarity Metric	RMSE	MAE
Pearson Correlation	0.9586	0.751
Cosine similarity	0.9645	0.7631
Jaccard Coefficient	**0.8656**	**0.7276**
MSD	0.9569	0.7548
SVD	0.9508	0.7492

Tab.4 Comparison of RMSE and MAE values for different similarity metrics

Tab.4 compares the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) values for different similarity metrics. The metrics compared are Pearson Correlation, Cosine Similarity, Jaccard Coefficient, MSD (Mean Squared Difference) and SVD (Singular Value Decomposition).

The highest values for both RMSE and MAE are observed for Cosine Similarity, followed by Pearson Correlation and MSD. The lowest values for both metrics are observed for Jaccard Coefficient and SVD. Hence, the metrics with higher RMSE and MAE values indicate a larger prediction error, while lower values indicate better accuracy.

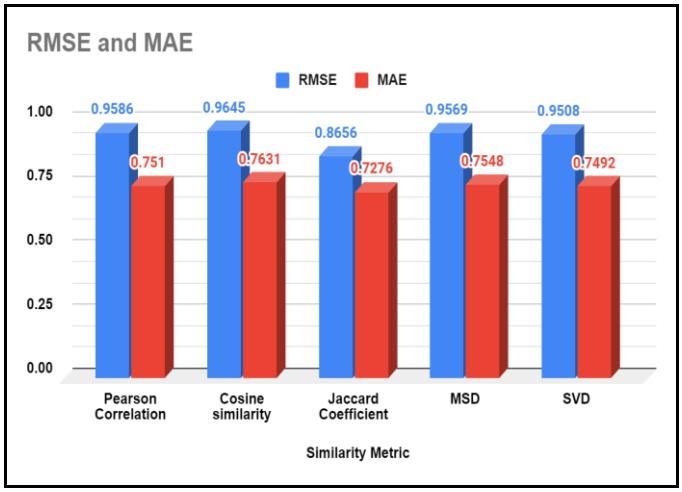


Fig.11 Shows the bar chart comparing RMSE and MAE values of different similarity metrics.

Fig.11 displays a bar chart showing Cosine Similarity's highest values for RMSE and MAE. The Jaccard Coefficient has the lowest RMSE and MAE values, indicating that it is the most accurate metric for predicting user-item ratings. However, the differences between the metrics are relatively small.

Some of the Advantages of the Jaccard index:

The Jaccard index has several advantages, including

1) It can make the recommendation system more reliable by providing accurate similarity scores between items or users, which can then be used to make accurate predictions.

2) It increases efficiency by reducing the number of comparisons to calculate the similarity scores because the Jaccard index only considers the intersection and union of two sets, which can be computed efficiently.

3) Execution is faster because it involves simple set operations, which can be computed quickly.

4) It gives accurate results when used appropriately, especially for sparse data sets with missing ratings.

The Jaccard similarity coefficient performs well in the user-based collaborative filtering approach because it finds users with similar preferences. It can help to predict accurate ratings for non-rated items.

In the item-to-item collaborative filtering approach, the choice of similarity metric may not significantly impact the predicted values because the focus is on finding similar items rather than users. However, the Jaccard index can still be a useful metric in this context because it can help identify items that are frequently rated together, which can be used to make recommendations based on users' ratings of other items.

6.1 Comparison

Stitini et al. [31] The paper "Investigating Different Similarity Metrics Used in Various Recommender Systems Types & Scenario Cases" explores the performance of different similarity metrics. The authors conduct an empirical study on four types of recommender systems: user-to-user, item-to-item, content based, and hybrid. They compare the performance of eight similarity metrics: Pearson, Spearman, Cosine, Jaccard, Adjusted cosine, Euclidean, Manhattan and Mean Squared Distance (MSD).

The study finds that the performance of the similarity metrics varies for different types of recommender systems and scenario cases. For user-to-user and item-to-item recommender systems, Cosine and Jaccard similarity metrics perform better than the other three metrics. For content-based recommender systems, Pearson correlation and Cosine similarity perform the best. In hybrid recommender systems, the choice of similarity metric depends on the specific scenario case.

Earlier studies on recommender systems compared various similarity metrics and found that the cosine and Jaccard metrics performed better than others in user-based recommender systems. This current paper builds upon that previous work and concludes that the model's performance is optimal when using the Jaccard coefficient similarity metric. Based on RMSE (**0.8656**) and MAE (**0.7276**) values, the Jaccard metric is shown to predict recommendations more accurately than all other similarity metrics.

7. Conclusion & Future Scope

7.1 Conclusion:

Recommender/Recommendation systems are becoming an essential tool for E-commerce on the web. They help users to find items they like and increase sales for businesses. With the massive user data volume, recommender systems need new technologies to improve scalability.

This study presented and evaluated various similarity metrics like Pearson Correlation, Cosine Similarity, Jaccard Coefficient, MSD (Mean Squared Difference), Sorensen Dice Coefficient and SVD (Singular Value Decomposition) for User-to-user collaborative filtering recommender systems. The results show that the Jaccard index can perform well for large data sets while producing high-quality recommendations.

Hence, the Jaccard index is useful when the data is sparse and missing values are common. Its advantages include improved reliability, increased efficiency, faster execution, and accurate results.

7.2 Future Scope:

1) Enhance the security measures to prevent fake ratings or user manipulation.
2) Enhance the evaluation approach for scenarios where there are no ratings available.

- Adopt proactive recommendation systems.
- Utilize privacy-preserving recommendation systems.

3) Implement a Deep Neural Networks recommendation system that provides dynamic results/recommendations.

8. REFERENCES

[1] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of CSCW '94, Chapel Hill, NC.

[2] Breese, J.S., Heckerman, D., Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence.

[3] Herlocker, J., Konstan, J.A., Terveen, L., Riedl, J. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22.

[4] Deshpande, M., and Karypis, G. (2004). Item-based top-n recommendation algorithms. ACM Transactions on Information System (TOIS), 22(1),143–177.

[5] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295).

[6] Fkih, F. (2021). Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. Journal of King Saud University-

Computer and Information Sciences, 33(4), 431-449. https://www.sciencedirect.com/science/article/pii/S1319157821002652?via%3Dihub

[7] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," Communications of ACM, vol. 35, no. 12, pp. 61–70, 1992.

[8] Manos Papagelis and Dimitris Plexousakis "Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents" Engineering Applications of Artificial Intelligence 18 (2005) 781–789 www.elsevier.com/locate/engappai

[9] Bell, R., & Koren, Y. (2007). Scalable Collaborative Filtering with Jointly Derived Neighbourhood Interpolation Weights. IEEE International Conference on Data Mining (ICDM'07), pp. 43–52.

[10] Saranya, K. G., Sudha Sadasivam, G., & Chandralekha, M. (2016). Performance Comparison of Different Similarity Measures for Collaborative Filtering Technique. Indian Journal of Science and Technology, 9(29), 1-7. DOI: 10.17485/ijst/2016/v9i29/91060

[11] A Case-Based Recommendation Approach for Market Basket Data Anna Gatzioura and Miquel Snchez-Marr IEEE INTELLIGENT SYSTEMS 2015.

[12] Recommender Systems: An overview of different approaches to recommendations Kunal Shah, Akshay Kumar Salunke, Saurabh Dongare, Kisandas Antala SIT, Lonavala India 2017

[13] Recommendation analysis on Item-based and User-based Collaborative Filtering Garima Gupta, Rahul Katarya, India

[14] Using collaborative filtering to weave an information Tapestry D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, Communications of the ACM, vol. 35, no. 12, pp. 6170,1992

[15] Recommender systems, Handbook, Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor. Springer 2010.

[16] Zhao, Zhi-Dan, and Ming-Sheng Shang. "User-based collaborative filtering recommendation algorithms on Hadoop." In 2010 Third International Conference on Knowledge Discovery and Data Mining, pp. 478-481. IEEE, 2010

[17] P. W. Yau and A. Tomlinson, "Towards Privacy in a Context Aware Social Network Based Recommendation System," Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, Boston, MA,2011, pp. 862-865. Doi:10.1109/PASSAT/SocialCom.2011.87

[18] Gao, Min, Zhongfu Wu, and Feng Jiang. "User rank for item-based collaborative filtering recommendation." Information Processing Letters 111, no. 9 (2011): 440-446.

[19] Grcar, M., Fortuna, B., Mladenic, D., Grobelnik, M.: k-NN versus SVM in the collaborative 3604iltering framework. Data Science and Classification pp. 251260 (2006).

[20] Hofmann, Collaborative filtering via Gaussian probabilistic latent semantic analysis. In: SIGIR 03: Proc.Of the 26th Annual Int. ACM SIGIR Conf. On Research and Development in Information Retrieval, pp. 259266. ACM, New York, NY, USA (2003).

[21] Bell, R., Koren, Y., Volinsky, C.: Modeling relationships at multiple scales to improve the accuracy of large recommender systems. In: KDD 07: Proc. Of the 13th ACM SIGKDD Int. Conf. On Knowledge Discovery and Data Mining, pp. 95104. ACM, New York, NY, USA (2007)

[22] Wikipedia link https://en.wikipedia.org/wiki/Collaborative_filtering

[23] Recommender Systems – The Textbook | Charu C.Aggarwal | Springer. Springer. 2016. ISBN 9783319296579.

[24] "A Study of Hybrid Recommendation Algorithm Based On User" Junrui Yang1, Cai Yang2, Xiaowei Hu3 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics

[25] Gomez-Uribe, Carlos A.; Hunt, Neil (28 December 2015). "The Netflix Recommender System". ACM Transactions on Management Information Systems. 6(4): 1–19. doi:10.1145/2843948

[26] A Study of Hybrid Recommendation Algorithm Based On User Xian University of Science and Technology Xian, China

[27] Crowe, N. (n.d.). Absolute bounds on set intersection and union sizes. Retrieved March 8, 2023, from https://faculty.nps.edu/ncrowe/intersect2.htm

[28] Recommender Systems in E-Commerce J. Ben Schafer, Joseph Konstan, John Riedl GroupLens Research Project Department of Computer Science and Engineering University of Minnesota Minneapolis, MN 55455 1-612-625-4002

[29] Rajeev Kumar, Guru Basava, Felicita Furtado, "An Efficient Content, Collaborative – Based and Hybrid Approach for Movie Recommendation Engine" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456- 6470, Volume-4 | Issue-3, April 2020, pp.894-904, URL: www.ijtsrd.com/papers/ijtsrd30737

[30] Maddali Surendra Prasad Babu, and Boddu Raja Sarath Kumar. An Implementation of the User-based Collaborative Filtering Algorithm / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3), 2011, 1283-1286

[31] Stitini, O., Kaloun, S., & Bencharef, O. (2022). Investigating Different Similarity Metrics Used in Various Recommender Systems Types: Scenario Cases. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-4/W3-2022, 327-334. doi:10.5194/isprs-archives-XLVIII-4-W3-2022-327-2022.

9. AUTHOR'S PROFILES

Kopanathi Sonali currently pursuing M.Tech- Data Science (CSE) from GITAM Institute of Technology, GITAM (Deemed to be University), Visakhapatnam. Her areas of research work are machine learning and deep learning. Her areas of interest are Recommended Systems, Neural Networks and data science.

Dr. S.V.G. REDDY completed M-Tech (CST) from Andhra University and has obtained a PhD in Computer Science and Engineering from JNTU Kakinada. He is an Associate Professor, the Department of CSE, GIT, GITAM University. His area of research work is data mining, machine learning and deep neural networks. He has guided various B. Tech and M. Tech projects and has publications in several journals. His areas of interest are drug discovery, computer vision, brain-computer interface, climate change and waste management.

K. Thammi Reddy completed his PhD (computer science and engineering) during 2008 from JNTU Hyderabad. He is working as Professor, Department of CSE, GIT and serving as Director, IQAC, GITAM University. His area of research work is data mining, machine learning and cloud computing with Hadoop. He has guided various B. Tech, M. Tech projects and had publications in several reputed journals and conferences.

V. Valli Kumari completed her PhD (CSSE) from Andhra University during 2006. She is working as professor, Department of CSSE, college of engineering, Andhra University. Her area of interest is Software engineering, Network Security and Cryptography, Privacy issues in Data Mining and Web Technologies. She has guided various B.Tech, M.Tech projects and had publications in several reputed journals and conferences. She received best researcher award and other various awards in the fields of teaching and research. She has undergone and completed various research projects.