ADAPTING A RECOMMENDATION SYSTEM ACCORDING TO CHANGING USER CONSIDERATION

Jehan Kadhim Shareef AL-SAFI

PhD Dissertation

Department of Computer Engineering

Program in Computer Engineering

Supervisor: Prof. Dr. CİHAN KALELİ

Eskişehir

Anadolu University

Institute of Graduate Programs

August 2021

ABSTRACT

ADAPTING A RECOMMENDATION SYSTEM ACCORDING TO CHANGING USER CONSIDERATION

Jehan Kadhim Shareef AL-SAFI

Department of Computer Engineering

Program in Computer Engineering

Anadolu University, Institute of Graduate Programs, August 2021

Supervisor: Prof. Dr. CİHAN KALELİ

The recommendation system is used to provide personalized recommendations to clients when they are selecting a product from a list of available options. The most often used technique for recommendation is collaborative filtering. In a collaborative filtering algorithm, the similarity factor employed in discovering the users with the same actions is one of the critical components of recommendations based on user rating for items; it does not consider any other information related to the recommender system entity (user and item). Although these methods can produce successful predictions, they have some drawbacks, such as scalability issues, finding suitable neighbors, and facing the cold-starting problem. Thus, the similarity computed in this manner is volatile for user pairs sharing a limited set of experiences and is undecidable for most users due to the absence of specified common tastes. In this dissertation, different solutions are presented to identify fluctuating user preferences and provide accurate recommendations.

The first solution proposes a novel neighbor selection approach based on correlation and slope that focuses on determining the relevance of entities. It is outlined the connected entities' similarities according to importance degree. Additionally, we proposed a new model for rating prediction based on the first accessible rating from nearby neighbors.

The second solution introduces a new user similarity measure to promote recommendation accuracy by determining each user's similarities even when only a few ratings are available. The suggested model considers only users' common ratings based on the proportion of dissimilar ratings and the highest value of this dissimilarity.

The third method provides a user similarity algorithm that uses item genre characteristics to provide more accurate suggestions. It establishes a relationship between users based on item genre preferences, identifies the current users' nearest neighbors in each genre, and generates a final list of nearest neighbors who share the same taste across all genres. Finally, it uses a definite prediction process to forecast ratings for goods. Additional to suggest a solution to the user cold-start problem via the "Ask-to-rate" technique/adaptive approach of inquiring about the new user's chosen item genre. Then, top users from the same genre are considered nearest neighbors for the active user. To assess the model's accuracy, we suggest two additional assessment criteria: the rating level and the rating level's reliability among users.

Numerous tests on real-world data sets have been undertaken. The experimental results demonstrate that the suggested algorithms generate highly accurate and reliable predictions.

Keywords: Collaborative filtering, Item-genre, Local similarity, Side information, Similarity measure.

ÖZET

BİR ÖNERİ SİSTEMİNİ DEĞİŞEN KULLANICI DÜŞÜNCESİNE GÖRE UYARLAMAK

Jehan Kadhim Shareef AL-SAFI

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Bilim Dalı

Anadolu Üniversitesi, Lisansüstü Eğitim Enstitüsü, Ağustos 2021

Danışman: Prof. Dr. CİHAN KALELİ

Öneri sistemi, müşterilere mevcut seçenekler listesinden bir ürün seçerken kişiselleştirilmiş öneriler sunmak için kullanılmaktadır. Tavsiye için en sık kullanılan teknik işbirlikçi filtrelemedir. Bir işbirlikçi filtreleme algoritmasında, aynı eylemlere sahip kullanıcıları keşfetmede kullanılan benzerlik faktörü kullanıcı derecelendirmesine dayalı önerilerin kritik bileşenlerinden biridir ve tavsiye sistemi başka bir varlık ile ilgili hiçbir bilgiyi dikkate alınmamaktadır. Bu yöntemler başarılı tahminler üretebilse de, ölçeklenebilirlik sorunları, uygun komşuları bulma ve soğuk başlatma problemiyle yüzleşme gibi bazı dezavantajları bulunmaktadır. Bu nedenle, bu şekilde hesaplanan benzerlik, sınırlı bir dizi deneyimi paylaşan kullanıcı çiftleri için değişkendir ve belirli ortak beğenilerin olmaması nedeniyle çoğu kullanıcı için karar verilemez. Bu tez çalışmasında, değişken kullanıcı tercihlerini belirlemek ve doğru önerilerde bulunmak için farklı çözümler sunulmaktadır.

İlk çözüm ile varlıkların benzerlik düzeyini belirlemeye odaklanan korelasyon ve eğime dayalı yeni bir komşu seçim yaklaşımı önerilmektedir. Bağlantılı varlıkların önem derecesine göre benzerlikleri ana hatlarıyla belirtilmiştir. Ek olarak, yakın komşulardan ilk erişilebilir derecelendirmeye dayalı tahmin üretmek için yeni bir model önerilmektedir.

İkinci çözüm olarak yalnızca birkaç derecelendirme mevcut olduğunda bile her kullanıcının benzerliklerini belirleyerek öneri doğruluğunu artırmak için yeni bir kullanıcı benzerliği ölçüsü sunulmaktadır. Önerilen modelde, farklı derecelendirmelerin oranına ve bu farklılığın en yüksek değerine dayalı olarak yalnızca kullanıcıların ortak derecelendirmelerini dikkate alınmaktadır.

Üçüncü yöntem ise daha doğru öneriler sağlamak için öğe türü özelliklerini kullanan bir kullanıcı benzerlik algoritması sunmaktadır. Öğe türü tercihlerine dayalı olarak kullanıcılar arasında bir ilişki kurar, mevcut kullanıcıların her türdeki en yakın komşularını tanımlar ve tüm türlerde aynı zevki paylaşan en yakın komşuların nihai listesini oluşturur. Son olarak, ürün derecelendirmeleri tahmin etmek için kesin bir tahmin süreci kullanılır. Ek olarak, yeni kullanıcının seçtiği öğe türü hakkında sorgulamaya yönelik "Derecelendirme İsteme" tekniği/uyarlanabilir yaklaşım yoluyla kullanıcı soğuk başlatma sorununa bir çözüm önerilmektedir. Ardından, aynı türden en iyi kullanıcılar, aktif kullanıcı için en yakın komşular olarak kabul edilir. Modelin doğruluğunu değerlendirmek için derecelendirme seviyesi ve derecelendirme seviyesinin kullanıcılar arasındaki güvenilirliği olarak iki ek değerlendirme kriteri önerilmektedir.

Gerçek dünya veri setleri üzerinde çok sayıda test yapılmıştır. Deneysel sonuçlar, önerilen algoritmaların oldukça doğru ve güvenilir tahminler ürettiğini göstermektedir.

Anahtar Sözcükler: İşbirlikçi Filtreleme, Öğe-Tür, Yerel benzerlik, Yan Bilgi, Benzerlik Ölçüsü.

ACKNOWLEDGEMENTS

First, I want to express my heartfelt appreciation to my supervisor, Prof. Dr. Cihan KALELİ, for believing in me and allowing me to pursue my research interests. His encouragement, direction, and support all contributed to shaping this dissertation. It is a tremendous privilege for me to be capable of working under his supervision.

Second, I would like to show my great respect to the jury committee. Thank you for accepting to be on my jury.

Third, I want to convey my gratitude and appreciation to the Turkish Scholarships Authority for their support throughout my Ph.D. journey.

Fourth, all my love and appreciation to my family for their care at all levels in my life. Words cannot describe your value in my life.

Fifth, all the affection and appreciation for the professors and friends in Iraq and Turkey; I acquired my strength and optimism with their assistance and encouragement.

Finally, thank you to everyone who supported me; thank you from the heart

Jehan Kadhim Shareef AL-SAFI

26/08/2021

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Anadolu University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Jehan Kadhim Shareef AL-SAFI

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

μ	:	The mean of all dataset ratings
μ_i	:	The mean rating of item i
μ_u	:	The mean of all ratings given by u
a	:	Active user
a(kNN, g1)	:	It is the K-NN for a in the first common genre; g1
a(kNN, gj)	:	The K- NN for a in the j^{th} common genre
ACOS	:	Adjusted Cosine Similarity
AE	:	Absolute Error
a_{NNf}	:	It is the final a′s NN
ap	:	The intercept point of X and Y
AVG	:	Average level
b	:	The slope or coefficient
BPM	:	Bayesian Posterior Mean method
C(a)	:	The average regression coefficient of a' ratings on the items' rating weight in the regression model
CB	:	Content-Based
CF	:	Collaborative Filtering
$Corr_a$	:	The set of correlated users with a
COS	:	Cosine Similarity
CPCC	:	Constrained Pearson Correlation Coefficient
DB	:	Demographic-Based
Diff	:	Difference
e	:	The error
g	:	Genre
H	:	An experimental value
I	:	The set of the common rating items
Index v_{gj}	:	The index of column vectors for each v_{gj}
int	:	Integer
I_u	:	Items rated by u

KB	:	Knowledge-Based
K-NN	:	Size of Nearest Neighbors
K-NNa	:	The K Nearest Neighbor set of a
M	:	The set of all users in the data set
M1	:	Model 1
MAE	:	Mean Absolute Error
$\min(U_i)$	:	The minimal number of ratings necessary to include i in the specified dataset
$\min(I_u)$	:	The minimum wanted ratings from u to be involved in the dataset
ML	:	Movie Lens data set
MSE	:	Mean square error
N	:	The set of all items in the data set
NEG	:	Negative level
NN	:	Nearest neighbors
NNfa	:	The final NN for a
P	:	Predicted rating
P (a, pi)	:	The a′s predicted rating value for unrated item pi.
P. L.	:	Predicted rating Level
PCC	:	Pearson Correlation Coefficient
Pi	:	Item
POS	:	Positive level
r	:	The actual rating
$\bar{r}a$	:	The mean rating for a
$\bar{r}u$	:	The mean rating of u
$r(i,j)_{NxM}$	:	Rating matrix
r(u, p)	:	The rating of item p by u
R^2	:	R-Squared a mathematical pointer
RDLD	:	Rating Dissimilarity and largest Difference similarity measure
Rel. L.	:	Reliability level
r_{La}	:	The rating level of a

RMSE	:	Root Mean Squared Error
RS	:	Recommendation Systems
S	:	The song in data set
sim(a, u)	:	The similarity between a and u
$sim(u, v)^{RDLD}$	:	Similarity between u and v according to RDLD model
Sim1 (a, u)	:	The PCC-based similarity between a and u
Sim2 (a, u)	:	The regression-based similarity between a and u
Spar	:	Sparsity
SPCC	:	Sigmoid Pearson Correlation Coefficient
SRS	:	Sequential Recommender System
St		star
SVD	:	Singular value decomposition
U	:	User
UGSM	:	User-Genre Similarity Model
Ui	:	The set of all users that rated item i
un	:	The Unknown genre
vgj	:	Column Vectors for each gj
w avr.min.sim	:	The weight of average minimum similarity between a
w_{LD} (u,v)	:	The weight of the largest difference between common dissimilar ratings
WPCC	:	Weighted Pearson Correlation Coefficient
w_{RD} (u,v)	:	The Dissimilarity Weight of common rating between u and user v
X	:	The independent variable (the users' and item' ratings in our present case)
Y	:	The measurable variable score

1. INTRODUCTION

Amidst increasing the websites, e-commerce sites, the number of online shoppers and goods has significantly risen. As a competitive operation, online shops require purchasing tools which can improve sales businesses, advantages, and client gratification. Recommendation systems (RS) assist these markets by offering recommendations for buyers according to their preference history. Therefore, recommendation algorithms are frequently associated with different expertise sectors to enhance RSs production and accuracy (Linden, Smith, and York 2003).

The RS uses approaches to filter data and present items relevant to the user's choices (Kortenhof 2017)(Beel et al. 2016). In general, RS requires data mining techniques to manage the growing volume of data, prediction algorithms to forecast the expected evaluation of active users for a particular item, and recommendation algorithms to deliver recommendations to the active user (Isinkaye, Folajimi, and Ojokoh 2015).

The RSs personalize websites. The RSs are essential applications that can help users choose their preferred product from a vast range of products (Ricci, Rokach, and Shapira 2014). The demand for new and more reliable RSs has increased since the websites, users, and various products and services increased.

Recommendation systems recognize users' interests in advance and use this knowledge to suggest items that the user may not have previously known them. As a result, recommendation systems have become essential for accessing information in e-commerce operations, reducing vast amounts of data, and directing users to meet their needs better (Beel et al. 2016)(Burke 2002).

To provide the appropriate content at the right time to the user, the RS collects the user's previous preferences, which can be an explicit rating for the items or an implicit rating. The system can deduce the implicit rating from the user's behavior towards a specific product. For example, the user took a long time on a particular product page; this reflects the user's interest in the intended outcome. Recommendation techniques can vary depending on the availability of updated information about user preferences and items. Accordingly, researchers in (Yang et al. 2016) have classified the RS technologies into collaborative filtering (CF), Content-based (CB), Demographic-based (DB), Knowledge-based (KB), and hybrid algorithms, as shown in Figure 1.1.

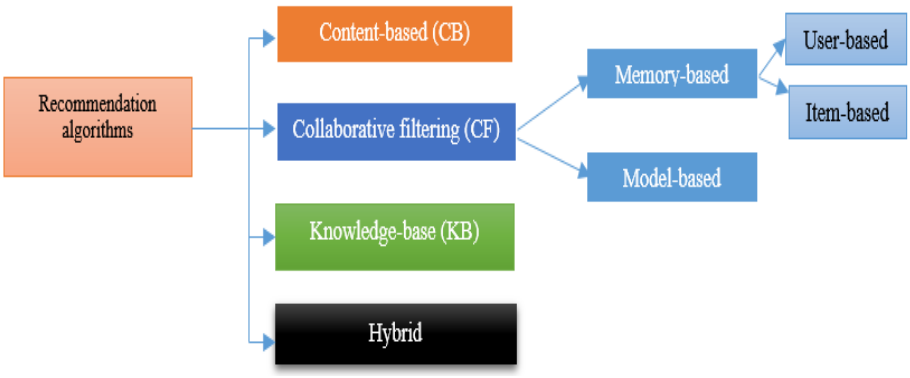

Figure 1.1. *Current recommendation algorithms and categories*

The CF algorithm recommends products preferred by users who are similar to the active user. The CB algorithm proposes products identical to those favored by the active user in the past by analyzing product ratings and contents. In a DB algorithm, user demographic information is exploited to generate recommendations. Finally, the KB algorithm depends on analysis and domain-specific knowledge of the product to be proposed. In addition to hybrid algorithms, which combine more than one of the above algorithms to overcome each algorithm's defects (Burke. 2007).

The most common methods for recommending items to users are CF (Breese, Heckerman, and Kadie 1998). CF systems measure the similar mindset of users by comparing their previous preferences. This degree of similarity is then invested in selecting the group of users who influence the final recommendations. Therefore, the similarity calculation is an essential step in the CF algorithm.

In recommendation systems, the number of available ratings is small compared to the number of ratings to be predicted. Therefore, it is necessary to obtain an accurate prediction based on general information. This process represents the main problem of traditional similarity measures because they rely only on local information in calculating similarity. As a result, they infer similarity from the evidence that might be insufficient in most cases and fail to find the actual convergence between users (Gediminas Adomavicius and Tuzhilin 2005).

Indirect or general similarity measures endeavor to defeat the local similarity measures problems by exploiting the actual or social connections to calculate the similarity between users (Luo et al. 2008)(Dell'Amico and Capra 2008).

Global similarity metrics improve the quality of recommendations and accurately predict user preferences even if the data is dispersed. However, there are methods concerned with prediction algorithms that take advantage of general and local similarity measures to ensure more accurate recommendations in most circumstances (Luo et al. 2008). A detailed overview of recommender systems is presented in Chapter 2.

1.1. Collaborative Filtering

The CF systems are based on our daily practice when we consult friends to make a specific decision about what movie to watch, what hotel to select, and so on. However, instead of the biased opinions of friends, the CF system harnesses the views of thousands of system users to find suitable recommendations for us. Furthermore, the CF assumes that user preferences are stable, is based on the theory that people who have agreed in the past tend to settle in the future.

Many popular recommendation apps have relied on the CF; Such as Ringo, Amazon, Movie Lens, etc., because it is the most popular among other recommendation techniques. However, the CF systems are classified into model-based and memory-based methods (Breese, Heckerman, and Kadie 1998). In model-based methods, statistical and machine learning methods discover the user model from the available data and generate recommendations. Various techniques are used in the model-based approaches, such as clustering (K. jae Kim and Ahn 2008), Bayesian networks (Breese, Heckerman, and Kadie 1998), decision trees, and neural networks (Billsus et al. 1998)(Paradarami, Bastian, and Wightman 2017).

The dimensions of the data are reduced by specific techniques and then rely on the acquired model to calculate the prediction for the required item (Ur Rehman, Hussain, and Hussain 2013). The model-base has several characteristics:

The ability to deal with data sparsity and scalability problems reduces the data dimensions, lowers the model implementation time, and uses the data efficiently through the used models (Das, Majumder, and Mali 2017)(Charu C. Aggarwal 2016). The prediction speed is faster than the memory-based technique since querying the data in the learned model is shorter than the time required to query from the entire dataset (Charu C. Aggarwal 2016). Despite its advantages, it suffers from the cost of building and updating the used models (Demissie 2018).

Memory-based algorithms are heuristics algorithms, which rely entirely on the user's rating record to infer prediction. These commonly include user-based and item-based methods. This technique involves finding similarities between user pairs to find the nearest neighbors (NN) and then prediction inferring. When defining a neighborhood, the required item rating is calculated based on the ratings of NN. This algorithm is the most popular and has been used extensively. The memory-based method provides relatively accurate results, but it needs more memory due to the continuous increase in items and internet users who need to generate thousands of recommendations per minute (Resnick et al. 1994)(Luo et al. 2008).

With proceed of RSs, researchers have condensed on exploiting the strengths of CF categories. Such as, authors in (Al-Shamri and Bharadwaj 2008) integrate memory-based and model-based methods to create a genetic approach that maintains accuracy with the memory-based process and avoids scalability by model-based approach.

Among the most important reasons that make memory-based models more common in creating recommendations are: predictive quality, simplicity and clarity of the algorithm structure, ease of justifying recommendations to users, and adding new data more flexibly than model-based methods. However, the process of creating requests for the active user "a" in memory-based forms consists of three main stages:

Similarity assessment: by estimating the similarity of the user with the rest users in the data set. Diverse methods have been presented for this purpose (Zeng et al. 2004) (Ahn. 2008) (Anand and Bharadwaj 2011)(Sun et al. 2012; Jesús Bobadilla, Ortega, and Hernando 2012) (Y. Wang et al. 2017).

Neighborhood formation: it is critical to select the proper subset of users based on their calculated similarity to the active user. The most frequently used strategy in this regard is the k-NN approach, which involves selecting the k most comparable users who have common items to participate in the suggestion process. A downside of this strategy is that users with relatively low levels of similarity may also contribute to the suggestion process, lowering the quality of the recommendations. Various approaches limit the closeness between a user and "a" to address this issue. If the similarity between the user and "a" is less than the threshold level, the user will not be in a's community. While this ensures that neighbor quality is not degraded, the number of neighbors may be restricted;

no users may meet the similarity requirement in some circumstances. Numerous ways recommend selecting all users who are favorably connected with "a" regardless of the degree of similarity (Chrsistian Desrosiers and Karypis 2008), mainly whenever the assessment range is sparse.

The NN CF algorithm utilizes information gathered about the use rating of this user to forecast interests (X. Su and Khoshgoftaar 2009). Using tools like PCC (Cremonesi, Koren, and Turrin 2010), vector space similarity (Ghauth and Abdullah 2009), and cosine-based similarity (Deshpande and Karypis 2004)(B. Sarwar et al. 2001), NN CF suggestions will routinely use the various measures of distance between users or things.

Regarding these two major options, here's how they are differentiated: the first kind, which relies on other similar-minded users to propose an item to a specific user, is the user-based approach. This model must locate the NN whose tastes or preferences are the same. Because of this, predictions of users who are active on the site and have unrated products rely on the rating of users with the same goods.

The second kind is the item-based model; this model identifies objects that are strongly linked or comparable to the target item and presents them to particular users. The item-based approach finds a group of similar things and predicts the user's rating based on the user's ratings for those items; this user's ratings for the target item's neighboring items constitute the prediction.

CF is concentrated on the k-NN algorithm to show users or products that have the highest probability of being suggested (Y. Park et al. 2015) (B. Sarwar et al. 2001) (Zeng et al. 2004).

Generating recommendations: it is produced by managing the collection of identified neighbors, combining their ratings for the specific item to predict the rating that the active user will assign the item.

The most often used method of aggregating ratings for item pi by active user "a" was presented by (Resnick et al. 1994) as Eq.(1.1):

$$P_{a,pi} = \bar{r}_a + \frac{\sum_{u \in NN(a)} \text{sim}_{a,u} * (r_{u,pi} - \bar{r}_u)}{\sum_{u \in NN(a)} |\text{sim}_{a,u}|} \qquad (1.1)$$

Where:

$P_{a,pi}$: The predicted rating for the active user a for item pi,

$\bar{r}_a$: The mean rating for a.

$sim_{a,u}$: The similarity between a and u, and

NN (a): The neighborhood of a.

Alternatively to this strategy, weighted median and weighted mode procedures may be used.

In (Garcin et al. 2009), the authors compare various aggregating approaches. Typically, the user is given a list of things rated according to their expected choice. Once all the ratings are anticipated, the system can provide suggestions in whichever format the application requires. While this fundamental analysis concentrated on user community generation, a comparable strategy from the item perspective is also pursued and has been widely employed with one of the most popular RS, Amazon (Linden, Smith, and York 2003). Item-based approaches begin by analyzing the user-item matrix to uncover relationships between different items and then use these associations to infer similarity between people (B. Sarwar et al. 2001). Ratings filtering by taste and quality are achievable due to dependence on people. In comparison, their independence from the content description enables them to be implemented in any area with access to important preference information. However, several obstacles have harmed the acceptance of CF systems, as described in the next section.

1.2. Obstacles of Collaborative Filtering

Sparsity: in CF, the sparsity matters relate to the shortage of available user ratings, making estimating closeness between specific user/item pairs impossible. This problem occurs because it is impossible to identify whether the user did not rate an item since he disliked it or does not get the opportunity to try it. Since each user's sample of reviewed things is a small percentage of the total pool of existing things, the overlap in ratings amongst two users, which is the basis for many similarity assessment methods, may be inadequate.

There are additional issues associated with the sparsity problem, including "cold start," which denotes a lack of information in the initialization step, preventing the selection of good recommendations.

Data sparsity intimates an insufficient number of ratings for items from users; sparseness in the <user x item> matrix, limiting the CF algorithm to pick the suitable set of similar users (Xue et al. 2005; Ahn 2006; Subramaniyaswamy and Logesh 2017). In parallel, the cold-start problem is a significant problem in recommender systems. This problem is divided into cold users and cold items. The problem with the cold user occurs by the lack of information about the user's preferences (Ahn 2006; Middleton, Alani, and Roure 2002)(G and A 2005). In addition, numerous efforts have been made until now to remedy the sparsity and cold-start problems. Some of them will be described in more detail in this dissertation.

Scalability: it is an issue that arises when the volume of information generated exceeds the capacity of users to handle it. It is challenging for memory-based CF, as all calculations are conducted at prediction time. As a result, a dimension reduction algorithm must deal with it and generate accurate recommendations in the shortest time and space.

Several of the most significant scalability challenges have been founded on model-based approaches, including dimensionality reduction. The authors in (Billsus et al. 1998) outline user ratings and rated items into a lower-dimensional space by singular value decomposition (SVD) of the original user rating matrix. The authors in (Bell et al. 2007) describe a system to improve the nearest neighbor approach by removing global impacts from rating data, hence increasing the comparability of entities and a method for simultaneously calculating resemblance weights for all pairs of entities. Apart from delivering superior recommendations to conventional similarity schemes, the technique is also highly time-efficient.

Privacy: the recommendation system collects and stores user data to compare and find similar users. It is stated that the depth or specificity of the user profile has a significant effect on the richness of recommendations.

However, a user may have a reasonable fear when disclosing personal information; sensitive data is possibly misused data for evil purposes. As a result, a trade-off occurs between user privacy and the quality of the recommendations generated (O'Connor and Herlocker 2001).

Vulnerability to attacks: collaborative recommender systems rely on their users' friendliness, i.e., there is an underlying assumption that users will participate with the system to obtain beneficial recommendations for themselves while also contributing valuable data for their neighbors.

A multitude of motives will make people interact with RSs. While these factors may contradict those of the system's owner or most of its users, they are supported by valid arguments (Ricci, Shapira, and Rokach 2015).

Malicious individuals may access the system with the apparent intention of manipulating user opinions in favor of or even against a specific item (Al-Shamri and Bharadwaj 2008). Numerous proposed attack-resistant CF techniques utilize trust scores collected from the user social network in addition to rating data to infer forecasts from people's grasp (Dell'Amico and Capra 2008).

1.3. Research Problem

This section explains the research problem that this dissertation will address. As previously mentioned, there are several drawbacks to the current recommendation systems. Although the current recommendation systems can attract users via recommendations, such systems need more knowledge about the user requirement in some instances. We examine whether knowledge about the user's actual interest, the weight of local similarity and dissimilarity for users, and side information of entities can fill part of the missing knowledge gap needed by recommendation systems. The details of each problem are clarified in the coming chapters.

1.4. Research Questions

1. Is it possible to identify the most promising nearest neighbors when defining the actual similarity between entities via determining the users' interests and importance, the weight of local similarity and contrast between users, and side information of entities, then creating the appropriate recommendations based on these similarities?

It is necessary to find similarity measures and algorithms that discover the actual preferences of users, most promising neighbors and create a recommendation system based on available information that provides accurate recommendations to answer this question.

2. Does the proposed recommendation system perform better than the previous recommendation systems in providing accurate recommendations?

To answer this question, we need to compare the performance of the proposed system with the traditional and the previous recommendation systems that depend on different methodologies in suggesting recommendations.

1.5. Dissertation Contributions

The dissertation makes the following significant contributions:

- Model of similarity based on PCC and regression: it provides a novel neighbor selection strategy based on correlation and slope to determine the entity's importance. It is summarized the similarities between the connected entities in terms of their relative importance. Additionally, we proposed a new model for rating prediction based on the first accessible rating from neighboring properties (J. K. S. Al-safi and Kaleli 2021).

- RDLD (Rating Dissimilarity and largest Difference) similarity measure: this model determines the degree of difference between user ratings to identifying users' similarity degree. The degree of dissimilarity is determined by calculating the weight of dissimilar ratings in co-rated items and the most substantial difference in co-rated items between users (J. Al-safi and Kaleli 2021).

- UGSM (User-Genre Similarity Model): by exploiting the side information. The suggested technique calculates users' average preference for each item genre as side information using a user similarity threshold value and discovers nearest neighbors. It identifies neighbors who share similar tastes across genres by intersecting user preferences across genres and treats them as the ultimate neighbors for the current user. Additionally, we propose a strategy for resolving the user cold-start problem using the "Ask-to-rate" technique/adaptive method. Furthermore, we propose two additional evaluation criteria to quantify accuracy: rating level and user reliability at that rating level (Al-Safi and Kaleli 2021).

1.6. Thesis Outline

The background of the dissertation is introduced in chapter 2. Chapter 3 described the PCC-slope- based similarity model. RDLD similarity model is explained in chapter 4. Entity side information is exploited to infer UGSM, which is introduced in chapter 5. Conclusion and future studies are presented in chapter 6.

2. BACKGROUND

There are dozens of television stations, catalog products, thousands of movies, and more, which are part of our everyday lives in our surroundings.

Users can utilize recommender systems to filter out items that are attractive, unique, or advantageous. They are distinguished from information retrieval systems in that they recover objects that match consumer behavior without requiring a query, hence facilitating "discovery" instead of "search." A recommender system's observation and learning of a user profile are similar to the queries presented by users in an information retrieval system.

The subsequent sections provide an overview of RS's various sorts of input and the forms of recommendations made. We then explore the many types of recommender systems and the multiple strategies for utilizing local similarity and entities side information for supporting RS.

2.1. Data Derivation

To create recommendations, the RS derives its data from the user rating matrix for products. This rating matrix represents all the ratings that users have given to the purchased or watched products. Ratings may be explicit or implicit. Explicit rating may be binary such as "like" or "dislike." Or it may be specific scores, such as the movie Lens, a Movie RS that allows numerical ratings from 0.5 to 5 (Harper and Konstan 2015). For example, "0.5" represents "No preference," and "5" illustrates the "Total preference" for the movie that the user has watched, or Yahoo! music dataset (http://research.yahoo.com) that allows numerical ratings from 1 to 100 to evaluate the music the user has listened (Pazzani 1999).

The systems derive users' implicit rating of products from their browsing behavior, allowing more information to be collected about actual user preferences. These ratings may be more intense than explicit ratings because they do not require user effort, but they are often less accurate. However, it is impossible to deduce the user's products because users only browse their favorite items. Accordingly, other types of information such as the item features, like movie genre, or the user's characteristics such as demographic, occupation, and age can determine the user's preferences (J. Ben Schafer et al. 2007)(Hasan and Roy 2019).

2.2. Types of recommender systems

Recommendation systems first came into operation in the 1990s and were limited to combining user inputs and providing recommendations to recipients. The first recommendation system called "Tapestry" followed this process. It is a messaging system that allows users to rate messages binary, either "good" or "bad."

The researchers called this system "collaborative filtering" (Goldberg et al. 1992). First, recommendations were made manually by users by specifying similar users with their tastes. Then, an automated selection process was introduced and suggesting similar users by analyzing usage patterns of all system users (Shardanand and Maes 1995)(Resnick et al. 1994).

The CF technique is based on automating "word-of-mouth" recommendations by gathering and recommending similar user views. The active user is the user for whom the system suggests the advice. Product features, user attributes, and feedback have been exploited either independently or mixed with CF to precisely predict user interests (Shardanand and Maes 1995). In addition to the CF algorithm described in detail in the previous chapter, we briefly present the other recommendation strategies in the following subsections.

2.2.1. Content-based filtering

The content-based system depends on the attributes of the items preferred by the user for creating the user profile. The characteristics of the favorites are compared with those available in the system to develop recommendations. The user profile varies depending on the learning methods used in the system (Burke 2002). For example, a restaurant recommendation system differs from a news recommendation system; each system has its characteristics and structure. Various methods such as decision trees and vector-based representations represent the user profile in this system (Pazzani and Billsus 2007).

This system suffers from multiple deficiencies that limited its success compared to the collaborative filtering system(Villegas et al. 2018). Among the shortcomings:

Require a detailed description of the item contents: It can be applied in areas that provide a detailed item description. For example, while it is capable of textual

description in articles, it lacks an accurate recommendation in media such as images or videos.

The lack of diversity in recommendations: users only rate their desire items, and the system cannot identify which undesired items.

The new user problem: The new user does not have any profile in the system, so it is difficult for the system to recommend what it prefers accurately.

2.2.2. Demographic filtration system

Demographic technologies are similar to collaborative links. Such as, the Lifestyle Finder system (Krulwich 1997) relies on demographic groups that include purchase history and survey response to generate recommendations. Creating a user profile depends on "interpersonal" connections and different types of users' data such as age, place of residence, occupation, and others. Users who are demographically suited to the active user are chosen to contribute to the recommendations.

Recommendations are generated based on the preferences of the active user's group. One of the essential features of this approach is that it does not require prior user ratings as in collaborative filtering and content-based technologies (Burke 2002). Despite this, it suffers from many problems, such as:

Data elicitation: The task of gathering demographic information is difficult because most users are undesired of disclosing their data, and incorrect data may exist (B. J. Schafer et al. 2007). Therefore, some systems have relied on importing demographic information from the users' homepages (Pazzani 1999).

Gray sheep problem: this problem occurs when some users have unique tastes and cannot belong to a similar group. Thus, it is challenging to generate a recommendation for such users (M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes 1999).

2.2.3. Knowledge-based system

This system is based on knowledge of the items and user preferences; considered a content-based system type. Items are recommended based on explicit item knowledge in a specific field that meet user preference. The difficulty of knowledge-based systems lies in the necessity to obtain a detailed knowledge base in advance. On the other hand, these systems do not suffer from the item or user cold-start problem or the data sparsity

problem. The collaborative filtering systems rely on user rating only face (Arekar, Sonar, and and Uke 2015) (Sielis, Tzanavari, and Papadopoulos 2014). There are two classes in knowledge-based system:

Case-Based recommender systems: in this category, the current case is compared with the possibilities available in the database and classified according to the degree of similarity between them. These systems provide explanations for why these elements are recommended.

Constrained-Based recommender systems: They depend on specific sets of preferences and are conditionally restricted.

2.2.4. Hybrid recommendation systems

Hybrid approaches work by combining the best features of multiple recommendation systems to improve the recommendations generated. The hybrid approaches in (Pazzani 1999)(Ricci, Rokach, and Shapira 2014) have been produced by integrating the CF with other recommender system strategies like content-based and demographic to provide a quality recommendation. For instance, the content-based recommendation technique can overcome the new user problem introduced by user-based CF and demographic filtering. The author in (Burke 2002) presents a comprehensive study of hybrid recommender systems.

2.3. Local Similarity

The similarity between users is a critical component of both user-based and item-based strategies in CF algorithms. The majority of similarity approaches estimate similarity solely based on the similarity of ratings between two users; we refer to this as 'Local similarity.'

The Pearson Correlation Coefficient (Resnick et al. 1994) and Cosine similarity (Breese, Heckerman, and Kadie 1998) methods are the most frequently used methods for determining the local similarity between users. Generally, resemblance computations are based on determining the similarity of rating vectors comprising user ratings of system items. Pearson correlation coefficient is used to determine the level to which there is a linear relationship between the ratings of two users (u and v) and is specified as Eq. (2.1):

$$sim(u,v)^{PCC} = \frac{\sum_{p\in I}(r_{u,p}-\bar{r}_u)(r_{v,p}-\bar{r}_v)}{\sqrt{\sum_{p\in I}(r_{u,p}-\bar{r}_u)^2}\cdot\sqrt{\sum_{p\in I}(r_{v,p}-\bar{r}_v)^2}} \qquad (2.1)$$

Where:

I: Denotes the set of the common rating items by user u and v.

$\overline{r}_u$, and $\overline{r}_v$: is the average rating value of u and v, respectively.

$r_{u,p}$, and $r_{v,p}$: denotes the rating of item p by u and v, respectively.

Another often used similarity metric for CF is the Cosine similarity (COS) (Breese, Heckerman, and Kadie 1998), as in Eq. (2.2):

$$sim(u,v)^{cos} = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\| \cdot \|\vec{r}_v\|} \qquad (2.2)$$

Where:

$\vec{r}_u$, and $\overline{\vec{r}_v}$: Is the vector of the user *u* and *v* rated, respectively. The magnitude of the vector is represented as a $/\!/\cdot/\!/$.

For item-based CF, amongst the most successful techniques for determining similarity is to utilize the Adjusted Cosine Similarity (ACS) (B. Sarwar et al. 2001), which determines the variance in the user rating system and is described as Eq.(2.3):

$$sim(u,v)^{Acos} = \frac{\sum_{p \in N} (r_{u,p} - \bar{r}_u)(r_{v,p} - \bar{r}_v)}{\sqrt{\sum_{p \in N} (r_{u,p} - \bar{r}_u)^2} \cdot \sqrt{\sum_{p \in n} (r_{v,p} - \bar{r}_v)^2}} \qquad (2.3)$$

Where:

N: represents the set of all items in the dataset.

The Spearman Rank Correlation is comparable to the Pearson Correlation, except that it is determined using the rank of items in the users' preference list rather than actual ratings (J. Herlocker, Konstan, and Riedl 2002).

Lathia et al., 2007 in (Lathia, Hailes, and Capra 2008) estimate the relationship between two users using concordance-based techniques. The degree of similarity between two users is defined by the amount of concordant, conflicting, and tied pairs of shared evaluations. When the desired data is binary, i.e., whether the user likes or dislikes an item, the Jaccard coefficient is used to compute the similarity proportionate to the number of common things between users.

Similarity calculated purely based on frequently rated products may not be a reliable reflection of the users' genuine affinity. Thus, weighing systems consider additional characteristics to weigh the similarities horizontally or vertically to make the resulting similarity better understandable.

For instance, (J. Herlocker, Konstan, and Riedl 2002) find that it is typical for active users to strongly link neighbors based on a limited number of common items. These neighbors were commonly found to be poor predictors of the active user because they were based on small samples (typically 3 to 5 co-rated items). Therefore, the more points matched between users, the more certain it is that the computed correlation accurately represents the actual correlation between the users. They propose a meaningful weighing technique that employs a linear drop-off for correlations with a sample size less than a specified threshold. In (Candillier, Frank, and Fessant 2008), the authors offer numerous weighted similarity measures for collaborative filtering based on users and items. According to the plan, the Jaccard similarity is used as a weighting system while other similarities are simultaneously promoted, such as the Pearson correlation coefficient. This method highlights how similar individuals are who like various items but who also post frequently. In (J. Herlocker, Konstan, and Riedl 2002), the focus of this approach is on products' co-rated and liked,' as opposed to 'co-rated.'

A different technique to rating objects suggests giving each object a score to be graded based on their remarkable capacity. A system that uses variance-weighting to assign weights to various items. According to how much variability there is in ratings for an item like an unpopular or popular item, such as a controversial topic, represents the actual degree of preference similarity far better than equal weighting methods.

The same method for rating domains in binary rating systems was outlined by (R. R. Liu et al. 2009), who recommended scaling ratings in comparing objects according to the amount of each thing that is being judged. This process is not the first time there have been various item weighting strategies. For example, inverse-user frequency (Breese, Heckerman, and Kadie 1998) has been used previously. However, as in (Jin, Chai, and Si 2004), points out an automatic item-weighting method is employed that, in effect, causes item vectors to be clustered in the item space. As a result, individuals with similar interests will get closer together, while those with disparate interests would be more remote from each other.

2.4. Side Information

Due to the RS's widespread application in various real-world contexts, recommender methods have been extensively explored. The RS handles the users, items, and characteristics. Social relations among users, connections between users, and things are all intertwined. The general view is that all infrastructure needs with an RS revolve around the three main components, the user, the item, and user-item interaction. Therefore, an RS's data is all associated with them. Data may be divided into two major categories: data on users and objects, such as clicks and ratings, and data on aspects of the users' profile and items' profile (Yue Shi, Larson, and Alan Hanjalic 2014).

Sequential interaction data and generic interaction data are two types of interaction data. As a result, the data from an RS are classified into three categories: (1) data on general interactions, (2) data on sequential interactions, and (3) data on side information.

The interactions between people and objects are typically depicted using an interaction matrix, in which each row represents a user, but every column represents a product. Each element in the matrix contains information about the sort of interaction that occurred.

The interaction data can be classified as explicit (i.e., users' ratings on a product) or implicit (i.e., clicking, purchasing, watching) based on the type of interaction (S. Zhang et al. 2019). Consequently, the recommendation objective is typically defined as a matrix completion task using this aggregated interaction data (M. Zhang and Chen 2020).

A sequential interaction data set is a set of sequences of user-item interactions within a certain period and sorted according to their date. Indeed, it can be classified into single-type and multi-type data sets based on the number of interaction events shown in the sequence. Multiple-type interactions, such as watching, and purchasing, frequently occur in practice (W. Wang et al. 2020).

A recommendation system based on sequentially interaction data is described as a Sequential Recommender System (SRS), which also uses a sequence of prior interactions to forecast the likely future interaction(s) (Quadrana, Jannach, and Cremonesi 2018) (S. Wang et al. 2019).

The data is frequently sparse (Hu et al. 2019); it is insufficient to capture users' preferences for items accurately. As a result, side information, such as attribute and social information, and external knowledge, are often used to address the sparsity affair.

User attributes (e.g., gender, age) and item attributes (e.g., genre, price) are the most common types of attribute information (Han et al. 2018). Typically, a user attribute's original data is stored as a user information table, with each row representing a single user and each column representing a single category. It is frequently paired with general or sequential interaction data to provide suggestions.

Social information is information on the social relationships that users have in common. In addition, external knowledge, such as item taxonomy and semantic relationships between concepts, about users and items, typically helps to understand better users' preferences and item features (H. Wang et al. 2018), hence increasing recommendation performance. External knowledge is primarily divided into two forms in RS: the item/user ontology and shared knowledge.

Typically, the ontology of users or items is represented as a hierarchical branching network containing the structural relationships between individuals or things. A type of ontology knowledge frequently used for recommendation purposes is item taxonomy information (Jin Huang et al. 2019). A tree graph of this type is utilized on Amazon.com, where the platform's products are organized using their category information.

By incorporating item ontology knowledge, it can be better understand consumers' multi-level preferences for items, increasing the explain ability of suggestions (Gao et al. 2019). However, propagating users' preferences across objects in the hierarchy tree network to extract multi-level selections remains challenging.

Because of entities' complexity and relationships, common knowledge is usually described as a heterogeneous and complicated network with various nodes and edges (Guo et al. 2020).

Incorporating common knowledge facilitates discovering and exploiting various external implicit relationships between users and boosting recommendation performance. However, it is still challenging to gather coherent and usable information between different entities via multiple links between them for good recommendations.

Recently, recommender systems have significantly grown. One of the valuable methods is the rating prediction method, which is extensively used to infer the future preferences of users.

Methods based on CF are a fundamental building block of many recommender systems. By identifying and exploiting patterns of similarity between users and products, the CF can forecast which items a user will favor (Breese, Heckerman, and Kadie 1998). However, with the rising availability of side information, information linked with objects such as product reviews, movie plots, and so on, there is considerable interest in utilizing such information to compensate for the ratings' sparsity (Y. Chen and De Rijke 2018).

Numerous techniques for incorporating side information into recommender systems have been explored. The majority of these techniques were created in response to the rating prediction problem. These techniques are based on latent factor models involving hybrid methods (Gantner et al. 2010).

The authors of (Singh and Gordon 2008) proposed a method for rating prediction using collective matrix factorization. This strategy includes both the user-item preference matrix and the item-feature content matrix into a similar latent space, allowing for exploiting both types of information through associated item factors.

In (Agarwal and Chen 2009), authors introduced regression-based models that use characteristics to estimate the latent factors. Their method determines the latent user and item factors by independent regression on user and item characteristics. The recommendation is generated using a technique based on the user and item factors.

Another study (De Campos et al. 2010) utilized Bayesian networks made of item nodes, user nodes, and item feature nodes to integrate content-based and collaborative filtering-based suggestions. During forecasts, content information is spread via feature nodes from purchased to non-purchased products, and purchase information is transmitted via item nodes from other users to the user of concern. The two components are then merged to generate recommendations. The study (L. Chen et al. 2021) focused on employing the item genres to raise predicted ratings' accuracy and overcome the sparsity problem. The algorithm involves creating a matrix containing the user's item weight by combining two vectors; the first vector is the user's taste vector, extracted from the user's ratings for items. While, the second vector is a vector that contains the degree

of items' belonging to a specific type, which results from the exploitation of the collective ratings of a particular item. Finally, the resulting matrix was used to get the prediction of the desired item.

2.5. Evaluation Metrics

RS that offers recommendations involves two sets of data: one for training and one for evaluation. It is anticipated that training ratings are present in the system, and people give test ratings in the future. System predictions are used to find out how accurate the algorithm is. The algorithm's accuracy is following the results of the forecasts concerning the test set ratings. We quantify prediction quality by utilizing a variety of different indicators, which are Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean square error (MSE).

2.5.1. Prediction accuracy metrics

We used many metrics to determine accuracy, one of which is Mean Absolute Error (MAE), the other is Root Mean Squared Error (RMSE) (Stephanie 2016), and finally, the Mean square error (MSE) (Willmott and Matsuura 2005) (Shardanand and Maes 1995). Measuring absolute divergence between expected and actual ratings of an item is one way MAE determines the bias in ratings. To calculate the mean absolute error, we can think of n as the set of items in the test set, think of p_i as the predicted rating for the i^{th} rating in the test set, and r_i : Represents the actual matching rating (B. J. Schafer et al. 2007), as Eq.(2.4):

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |r_i - p_i| \tag{2.4}$$

Where:

r : represents the actual rating;

p : represents predicted rating;

n : indicates the total number of ratings.

The RMSE is defined as accumulating errors and squaring the result; as shown below in Eq. (2.4):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (r_i - p_i)^2} \tag{2.5}$$

Unlike the MAE, which considers every user's differences to be equal weight, the RMSE focuses on the effects of significant errors. The root means square error (RMSE) represents the standard deviation between the expected and original rating values. Numerous errors are highly undesirable; the (Stephanie 2016) research results show that a minor variation in root mean square error (RMSE) leads to a significant increase in the accuracy.

MSE is a quality metric used to calculate the mean squared error between actual and projected ratings. As in Eq. (2.6):

$$MSE = \frac{1}{n} \sum_{i=1}^{n} |r_i - p_i|^2 \quad (2.6)$$

Thus, lower MAE, MSE, and RMSE values signify a more accurate model.

2.6. Literature Review

Numerous researches in the field of recommender systems are described in this section. They are classified into two categories based on the theories provided in this research.

- The first category is concerned with applying Collaborative Filtering techniques, such as similarity metrics and prediction algorithms, to improve recommendations accuracy by utilizing the local similarity of common ratings between entities, as follows:

Many researchers have offered various similarity measures. Even though there have been several differences, the common idea is to check similarity among users or items using some action to recommend items or users based on the highest similarity. CF is rapidly expanding and affects the majority of recommender systems. The K-NN algorithm is the adaption algorithm used in a memory-based CF recommendation method. The recommendation procedures of K-NN-based recommender systems are reliable and deliver correct suggestions with CF (Z. Lu and Shen 2015) (Borràs, Moreno, and Valls 2014). Researchers conducted numerous experiments on the suggestion process to discover more effective NN algorithms. (J. Herlocker, Konstan, and Riedl 2002) perform a study on the item selection of CF. According to their exhaustive investigation, PCC is a viable option for determining user correlations, and the optimal method for establishing a neighborhood is to hire the most comparable K users. The PCC measures the linear

correlation between two vectors of user ratings (X. Su and Khoshgoftaar 2009; B. Sarwar et al. 2001). However, the PCC similarity measure does not regard the number of common ratings.

Additionally, the Constrained PCC is proposed. It is a somewhat adjusted version of Pearson's correlation that recognizes only the pairs of ratings on the same side, e.g., both being positive or negative, by taking the median value in the rating scale in the dataset to gain a good correlation (D. E. Chen 2009). Spearman's Rank Correlation is also used as a similarity measure, where the similarity between two vectors is determined based on the similarity of ranks of values in these two vectors (J. L. Herlocker et al. 1999; J. Lu et al. 2010).

The cosine measure model studies the angle between two vectors of ratings; the smaller angle is considered a more remarkable similarity between these two vectors. ACOS is used in some item-based CF methods for similarity among items where the difference in each user's use of the rating scale is considered in the similarity calculation (X. Su and Khoshgoftaar 2009; B. Sarwar et al. 2001).

Even that, there are some weaknesses in the methods as well. The most common CF problems are data sparsity and cold-start. Data sparsity indicates low ratings, or sparseness, in the <user x item> matrix (Xue et al. 2005; Ahn 2006). In comparison, the cold-start problems relate to the difficulty of recommending new items to new users where there are not enough ratings available from them (Middleton, Alani, and Roure 2002; Ahn 2006).

The propped model concentrates on the cold-start problem for new users with few ratings and selecting proper nearest neighbors. Previous studies on the cold start problem focused on new entities with no rating record at all. These studies provide different recommendation systems that combine content information with rating data to solve the cold start problem (Z. Huang, Chen, and Zeng 2004; Q. Li and Kim 2003; S. T. Park et al. 2006). The content-based similarity is used for new entities with no rating record.

On the other hand, researchers propose various researches on creating more suitable K-NN in the recommendation process.

The authors of (T. H. Kim and Yang 2007) suggest an effective threshold-based technique for CF community selection. The model substituted neighbors for a client with unusual interests.

In (Baltrunas and Ricci 2008), the prediction accuracy improved in their suggested model since an adaptive neighborhood selection model based on user taste is proposed. This model takes into account dynamic changes in the user profile and the profile of a particular object. It calculated the similarity between individual users using a selection of co-rated things.

The purpose of the heuristic similitude metric in (Ahn. 2008) was to enhance recommendation efficiency during cold-start settings. In this study, a system called Proximity-Impact-Popularity (PIP) was presented that utilized collaborative filtering to analyze three user behavior variables. The Proximity factor has been used to indicate whether the two ratings agree or differ. The Impact factor considered the positive reputation given to an item by users as a proxy for similarity. The popularity metric took the space between users and their average rating into account.

Many researchers provide valuable confirmation on the growing achievement of CF methods. In (Baltrunas and Ricci 2008), the authors considered dynamic changes in user profiles, and they are adaptive neighborhood selection approach based on individuals' preferences for a given product. Thus, actual evidence supports the assertion that this technique improves the accuracy of recommendation forecasts.

Early methods to solve data sparsity integrated the weight of similarities in the existing similarity behavior. Typically, this weight is determined by the number of popular items among customers.

In (Koren 2010), the authors offer a model for determining user connections based on a portion of a dataset rather than the full dataset. They chose neighbors based on the variance of the procedure for computing user similarity. This model increased the recommendation accuracy. By combining the Jaccard index and mean squared differences, authors in (J. Bobadilla, Serradilla, and Bernal 2010) create a new similarity measure.

The study in (H. Chen, Li, and Hu 2016) investigates the rating variation; this study employs a balancing factor to combine the PCC and adjusted cosine similarity.

Wang Yong et al. (Y. Wang et al. 2017) suggested a nonlinear mixed similarity measure algorithm based on users' rating information; it reveres the similarity and user asymmetry. It strengthens the relationship between users and items, but it suffers from a high time complexity.

This technique (Soojung Lee 2017) is predicated on the premise that two consumers of more common goods will share a greater degree of similarity. In (Zhu et al. 2018), the extended Jaccard index concept was created and utilized to estimate similarity. While it has been demonstrated that sparsity may be mitigated by merging several common ratings using similarity models, the method does not incorporate contextual information from rating data.

Huang et al. (M. Huang, Wang, and Zhou 2019) aimed to mitigate the data sparsity using a linear regression model to fill unrated items in the user-item rating matrix and find the similarities between users. This model considers the average user ratings and the average item ratings as the features and the user's rating as the label. The model used the traditional collaborative filtering algorithm for rating prediction after filling the user-item rating matrix. This model can mitigate the data sparsity, find more suitable user neighbors, and gain more rating prediction accuracy.

He and Jin (X. He and Jin 2019) proposed a similarity measure model based on users' preferences for item attributes, commonly rated items, and the number of widely rated items. It develops the recommendation production's performance by setting strong relationships between users and items and mitigating the data sparseness. This model achieved better performance than the other tested methods in terms of accuracy and diversity.

Jaccard similarity (Koutrika 2009) is a similarity measure that overlooks the absolute values and the average value of users' ratings. A similarity measure model that considered the proportion between absolute rating values and the number of co-rated items was suggested by (Ayub et al. 2020) to improve Jaccard similarity measure performance. This model is based on users' rating preference behavior and implied a threshold on the average user rating value. This measure promotes prediction quality and recommendation accuracy.

Once similarities among users are available, the prediction of item rating by the active user can be estimated based on the average rating of the active user and the similarities with their neighbors as described in (Ahn. 2008) for the user-based CF algorithm.

- The second category examined strategies for increasing recommendation accuracy through the use of entity ratings and side information, as follows:

Essentially, the RS relied on the filtering algorithm to obtain entity information. Researchers proposed several similarity measures to CF-RSs, attempting to demonstrate the accuracy of the recommendations to the desired level. Using entity-side information is a common strategy for overcoming the RS's problems and obtaining suitable recommendations (C. Li et al. 2015).

In (Vincent Schickel-Zuber 2005), the model forecasts ratings based on a continuous scale; instead of classification, the model used the regression approach. The standard K-NN algorithm using PCC and Cosine as similarity measures was confronted with the SVM regression. The multiple regression methods had been used to estimate the compound attributes' weights. At the same time, ontologies calculated the model and estimate utility values.

The accuracy and quality of recommender systems diminish when entity information is insufficient (Ahn. 2008). As a result, several solutions to the sparseness and new user issues have been offered.

The suggested model in (Liang, Bo, and Jun 2009) investigates user correlations and NN determination using a portion of a dataset rather than the full dataset. They proposed a unique model in CF for choosing the neighbors of active users based on their similarity variance. This model enhanced the accuracy of the CF forecast.

Most CF models measure user comparison on the whole ratings and ordered user relations with active users to select the NNs. But we debate this is irregular since we consider that users have a similar partial interest, not on the whole rated items (Liang, Bo, and Jun 2009). Therefore, our researches are based on the part of the dataset selected according to specific conditions.

Item genres are incorporated into the RS to increase the RS's accuracy. Several studies have concentrated on utilizing the movie genre feature to collect user input and enhance suggestions, particularly for massive datasets (O'Donovan et al. 2008; O'donovan et al. 2009). Others improve an RS called "Taste Weights" (Bostandjiev, O'Donovan, and Höllerer 2012); analyzing a user's Facebook profile identifies their preferred music genre. Then, the semantic web resource "DB pedia" is utilized to discover new artists performing music in the same genre as the active user's favorite and recommend such music to other users.

In (Y. Zhang and Song 2009), the genre-based algorithm was developed using the notion of item genres rather than items themselves. The algorithm picked neighbors based on their genre preference, comparable to that of others in the community. This research used user data to determine item similarity, and CF sparsity is alleviated.

The authors in (Jesús Bobadilla et al. 2012) proposed implementing a neural learning-based user similarity measure that would deliver favorable recommendations to users with few ratings (2 to 20 ratings). Additionally, the experimental results show that this method is even more accurate, recallable, and repeatable than prior systems.

In (You et al. 2013) proposes a method for resolving the cold-start status using a clustering approach. This approach is constructed using the user's preferred item ratings and genres to determine similarity. The findings indicate that the proposed model improves the cold start solution considerably and outperforms comparable techniques in the recommendation and cold-start circumstances. As a result, this method outperforms specific existing recommendation algorithms in terms of accuracy.

The authors (Gogna and Majumdar 2015) enhance prediction accuracy by using auxiliary data on the user's demography and item characteristics. The model incorporates supervised label terms to make feasible recommendations for new entities using the matrix factorization framework.

The research conducted in (Rosli et al. 2015) is focused on developing a model that mitigates cold-start circumstances using genre interest and "Likes" and "Co-Like" movies data from Facebook Pages. CF cold-start issue experimentally solved by this model. However, new users may abandon the system due to flaws in the suggestions during the

initial phases when users lack the necessary ratings to give to the CF in RS to calculate the user similarity.

Authors in (G. Adomavicius et al. 2015) reasoned differently to increase the accuracy of RSs. Their approach is based on the addition of contextual information to establish relationships between users or products. As a result, they suggest many mathematical methods for information filtering and modeling that demonstrate through experimental results that contextual information is critical for receiving correct recommendations.

Authors (H. R. Zhang, Min, and Shi 2017) propose a regression-based three-way RS to minimize the mean cost of various RS actions. They predicted ratings using a regression-based binary recommendation system based on slope one (Lemire and Maclachlan 2005) and the k-NN algorithm (B. Sarwar et al. 2001).

In (Frémal and Lecron 2017), a clustering method using the item's genre information is presented. This model concentrates on establishing a hybrid approach, combining the CB clustering method and CF, instituted by the items genre's investment. The model improves the rating prediction operation with a less time-consuming model.

In (Jiang et al. 2018), the model merged trusting data with user similarity to enhance the slope one model and generate the recommendation formula. The study in (J. Li et al. 2018) introduces a hybrid approach for solving sparse data by determining user interest. The movie feature is used to create a vector of movie attributes, merged with the user rating vector to generate a user interest vector to calculate users' similarity.

The researchers employed trust data to boost prediction accuracy and circumvent cold start and sparsity issues. In (Yuan, Zahir, and Yang 2019), two approaches are presented for calculating user trust in items according to their ratings. Although it is challenging to obtain trust information between users and objects, the proposed method performed better than tested methods in addressing the intended problems.

In (Zare et al. 2019), the authors use location information in recommendation algorithms to create a mobile learning application focused on presenting the user with favorable recommendations.

Differently, in (Ai et al. 2019), the authors introduce a user similarity model based on genre preferences to present a weighted link prediction model according to a complex network modeling and the detected user community.

In (D. Wang, Yih, and Ventresca 2020), a new NN selection model that combines structural and rating-based similarity calculations has been suggested. The experimental findings confirm that the memory-based CF algorithms exceed the model-based CF algorithm in prediction accuracy. This model was achieved on a few neighbors comparing with the traditional memory-based CF models.

Li and Li (S. Li and Li 2020) constructed an algorithm based on user characteristics and user preferences. They used multiple similarity measures, like user score, user attribute, and user interest similarity, to solve the cold-start problem and enhance the recommendation accuracy. This model used even user registration information to obtain user similarities. The model can solve the mentioned problems.

The authors (Moscato, Picariello, and Sperli 2020) present a music recommendation system that uses users' behavioral features in the model and exploits the user's overall mood and sentiment to produce recommendations. In addition, they blend CB recommendation with users' personality qualities on social networking sites, behavior, interests, and requirements and examine several methods for inferring users' personality qualities.

Researchers continue to exploit the item genres to create a recommender system; in (Z. Su et al. 2020), novel user-based similarity measures based on a vector to hold the user similarity using items genre is presented. The model contains finding the global, local, and Meta similarities between users to create the similarity vector and reveal and estimate user associations in different scenarios. They emphasize that users' similarities might be defined differently depending on the various items genre.

Researchers in (Yadav, Duhan, and Bhatia 2020) involve overcoming a new user cold-start challenge, weak recommendations accuracy, and other RS' issues by establishing a user profile on Linked Open Data initiative, collaborative aspects, and networking site attributes. The item similarity is determined by finding the ontology between items; then, they calculated the similarity according to the user rating for items.

The study (L. Chen et al. 2021) focused on employing the item genres to raise predicted ratings' accuracy and overcome the sparsity problem. The algorithm involves creating a matrix containing the user's item weight by combining two vectors. The first vector is the user's taste vector, extracted from the user's ratings for items. The second vector is a vector that contains the degree of items' belonging to a specific type, which results from the exploitation of the collective ratings of a particular item. Finally, the resultant matrix was utilized to obtain the required item's forecast.

3. A NEIGHBOR SELECTION MODEL FOR RECOMMENDER SYSTEMS USING CORRELATION AND REGRESSION

The recommendation system is already being used for personalized recommendations to clients when selecting a product from a list of available goods. The most often used mechanism for recommendation systems is collaborative filtering. In a collaborative filtering process, the similarity ratio is one of the essential aspects of suggestions, independent of any other information about the entity. This chapter introduces a novel neighborhood selection methodology based on correlation and regression relationships quantifying the significance of entities' effects on one another. The presented method takes into consideration the connected entities' similarity in importance.

Additionally, we recommend a new rating prediction algorithm that considers the first accessible rating from the closest neighboring users. In complement to our proposed model, we also tested several novel models. We undertake three genuine experiments on three actual data sets to evaluate our model accuracy and compare with state-of-the-art similarity measures. It is shown that the suggested framework provides an increase in accuracy over the predictions of the approaches used previously. Using the knowledge on how important a neighborhood is for obtaining a suited recommendation system, we demonstrated that the recommendations found on this platform are accurate. Thus, our suggestion will be a successful technique of selecting the nearest neighbors, and it will assist the recommendation process by helping to filter potential candidates.

3.1. Introduction

Numerous studies are attempting to resolve data sparsity, improve recommendation accuracy, and develop the NN selection of CF algorithms. For example, the CF-boosted default vote compensates for the user's unrated ratings (X. Su and Khoshgoftaar 2009). On another side, some measures utilized contextual information to build a similarity model without considering the traditional similarity measures that do not depend on the user ratings or supplementing the missing rating data (H. Liu et al. 2014).

However, CF algorithms have several defects. Despite its effectiveness, the user-based CF has scalability and real-time performance constraints. The complexity of numerical procedures increases linearly with the number of customers, which in

conventional commercial applications might reach many millions. As a result, the user-based CF can't prove the accurate suggestion.

Additionally, CF is incapable of handling recommendations from individuals with different interests. When users' interests diverge, it makes bad suggestions (Badrul Sarwar et al. 2000). These restrictions cause the customer to have doubts about the RS. The suggested research focuses on such CF constraints and attempts to overcome them by using a novel neighborhood selection approach.

3.1.1. Problem definition

Predictions of the requested destination for the active user rely on the neighbor liking for the rated item in the CF algorithm (X. Su and Khoshgoftaar 2009). However, it is natural for an individual to have several interests. For instance, a person may enjoy both 'Fantasy' and 'Drama' film genres concurrently. If we anticipate the genre "Drama" using value expectations for the "Fantasy" genre, the prediction findings are suspicious.

3.1.2. The contribution

One may deduce the critical nature of picking areas with comparable content to obtain an accurate rating forecast from the constraints discussed before. If two correlated users have similar slope values, they can be considered suitable neighbors, increasing RS accuracy. To create a more accurate NN selection model, we combine the conventional similarity PCC-base relationship (Jin, Chai, and Si 2004) with the regression coefficient (slope) values of entities (Sarstedt and Mooi 2014).

The suggested model accounts for the influence of users' and items' rating behavior on the weights of items and users by utilizing the multiple regression approach on associated user ratings. It is based on the PCC and the slope for determining similarity.

Additionally, we proposed a novel model for rating prediction based on the first accessible rating from nearby neighbors. We evaluate our model's efficacy against a variety of contemporary CF methods using benchmark datasets.

To achieve a good balance between model sparsity, complexity, and computing time, we have determined to employ a subset of the dataset, users, and objects from the benchmark datasets to create our model. Our model uses a subset of the dataset, users, and items in the high rating numbers from the benchmarks to mitigate these concerns.

Several CF approaches have been applied to get at correlations for different people or items. The application of the PCC creates this proposed solution stated in Equation (2.1), K-NN algorithms given in subsection (1.1), and multiple linear regression below. The traditional similarity measures like adjusted COS and PCC are accurate, but these techniques are only applicable in specific scenarios (Jin, Chai, and Si 2004)(Gazdar and Hidri 2019); because they investigate only the linear correlations and ignore the nonlinear correlation entities' weight impact on those relationships. Thus, entities' weight influence might yield greater authenticity for the entities' commonalities.

However, in contrast to standard CF algorithms, which use rating similarities, CF models employing rating divergences are being investigated. Our approach uses correlation connections combined with regression coefficients (slope) for a neighbor's selection model (PCC-Slope-based model).

The suggested models outperform standard CF models in rating prediction accuracy, particularly on sparse data sets. The research's most significant contributions are as follows:

1- The average influence of user ratings on the weight of the item is computed, as is the moderate impact of item ratings on the weight of users.

2- A novel correlation and regression-based neighbor selection approach is provided for constructing a similarity model between entities.

3- We present a model for rating prediction based on dynamic K-NNs and a subset of a dataset to decrease the model's sparsity, runtime, and complexity.

4- We are increasing the accuracy of recommendations made by recommender systems.

3.2. Regression Method

When employing the linear regression approach, you may learn which characteristics are important in forecasting the outcome and find how various variables are connected. It uses X as an input and calculates Y as output depending on the connection between X, and Y. One independent variable can be written in regression as Eq. (3.1):

$$Y = ap + bX + e \tag{3.1}$$

Where:

Y: The score calculated based on weight measurements of both items and users,

ap: The point where X and Y intersect,

b: is depicts the slope,

X: In this situation is the independent variable (the users' and item's ratings), and

e: is an abbreviation for error.

The used datasets in this research contain many features. Thus, the suggested model uses multiple linear regression. So, the regression is done as in Eq. (3.2):

$$Y = ap + b_1X_1 + b_2X_2 + \cdots + b_kX_k + e \quad (3.2)$$

We have K different independent variables, and every variable has its slope, but there is one error. For K>2 independent variables, locating the b values (slopes) is complicated, so we need matrix algebra for calculations (Sarstedt and Mooi 2014).

3.3. The Proposed Model for Selecting Nearest Neighbor

The proposed similarity measure and a new model for rating prediction are demonstrated in this section.

3.3.1. The correlation-regression-based neighbor selection model

It is important to use the available information to gain the most useful information from the entities' profiles. Thus, it is necessary to investigate how the user rating influences the item weight and how item rating affects the importance of user opinion to produce a trustworthy neighbor selection model that boosts RS accuracy.

The entities' impacts are determined by applying the multiple regression techniques. Then, it follows that we urge computing the relevance of linked entities to harness them in similarity measures and find out who is an active user's NN.

The initial step in the suggested model is to compute the correlation connection between entities to derive the PCC, as shown in the following Eq. (3.3).

$$Sim_1(a,u) = corr\ (a,u) \quad (3.3)$$

Where:

The similarity between a and u is denoted by $Sim_1(a,u)$.

Then, the second component of similarity is determined using the multiple regression techniques. The value of the average regression coefficient $C\ (u)$ or slope denotes the significance of correlated entities. The second similarity between entities is represented by the absolute differences in the slope values of active users and correlated users, which result in the formation of NNs, as defined in Eq. (3.4).

$$Sim_2\ (a,u) = |C\ (a)\ - C\ (u)\ |,\ \ u\ \in\ corr_a \tag{3.4}$$

Where:

$Sim_2\ (a,u)$: express the similarity between a and u based on the average regression coefficient.

$C(a)$: denotes the average regression coefficient of a' ratings on the items' weights in the regression model, and similarly for u,

$corr_a$: This variable indicates the collection of users with whom a is correlated.

The regression technique is used to get the average regression coefficient $C(u)$ of users. The $Sim_2\ (a,u)$ Values are sorted ascendingly to determine which NNs are assigned to the active user.

Similarly, the proposed technique is used to compute item-based similarity metrics across item rating vectors. The user rating weight is utilized as the regression model's output to infer the item's importance based on user ratings, creating the NN for the current item.

The correlation between entities in $Sim_1(a,u)$ is in the range [0, 1], and the same is true for $Sim_2\ (a,u)$. Thus, comparing the similarity of entities is possible. The proposed procedural stages are depicted in Figure 3.1.

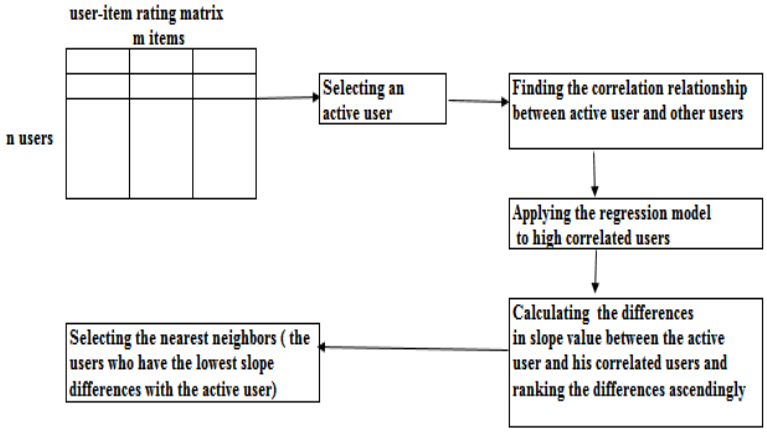

Figure 3.1. *The suggested PCC-regression-based NN model*

3.3.2. A novel technique for forecasting

Calculating the number of acceptable K neighbors is a key step in the K-NN method. If K is small, it may lose several comparable neighbors. While many K allows for selecting very different neighbors, this results in a decrease in the predictive accuracy of recommendations. Thus, changing the value of K may aid in prediction accuracy (Ricci, Shapira, and Rokach 2015)(Zeybek and KALELİ 2018). As a result, our approach utilizes a dynamic K value for NNs. A unique rating predictor based on the first accessible rating from NN has been suggested to acquire NNs and arrange them ascendingly according to the difference in the essential quality of a and his linked NNs.

After recognizing an active user's neighbors, one may anticipate the ratings of a to item pi based on the first accessible rating from his neighbors. Eq. (3.5) illustrates the formula for user-based model prediction.

$$P_{a,pi} = r_{u,pi} \quad \Leftrightarrow \quad u \in K - NN_a \text{ and } pi \in I_u \qquad (3.5)$$

Where:

$P_{a,pi}$: is the predicted rating value for an unrated item pi as determined by a.

$r_{u,pi}$: is the $u's$ rating value to pi.

$K - NN_a$: denotes the set of $a's$ k nearest neighbors.

I_u: denotes goods that have been rated by u.

Similarly, the technique is done to things rather than individuals in the item-based approach.

3.4. Experiments and Analysis

We conducted experiments on the frequently used data sets to determine the effectiveness of the proposed approach for finding NN and the proposed model for rating prediction in the user-based and item-based CF algorithms.

3.4.1. Data sets

We evaluated our model's performance using the Movie Lens 100k (ML 100k), Movie Lens 1M (ML 1M), and Movie Lens 20M (ML 20M) datasets (Harper and Konstan 2015). The Group Lens Research Group compiled this dataset at the University of Minnesota (http://www.grouplens.org) for the Movie Lens. At least twenty ratings are assigned to each user. The datasets are described in the following manner:

ML100k dataset: This data set contains 100,000 ratings from 943 people for 1,682 movie. The grading scales range from 1 (worst-case scenario) to 5 (the best case). This dataset is 93% sparse. The level of sparsity is determined by Eq. (3.6) (Veras De Sena Rosa et al. 2020):

$$Sparsity = 100 * \left(1 - \frac{\text{all observed ratings}}{m*n}\right) \tag{3.6}$$

Where n is the total number of items and m is the total number of users. Table 3.1 contains information about the remaining datasets. We studied the ML 20M dataset for our suggested model. Simultaneously, we utilized the other data sets to make comparisons to prior studies. St stands for star, U for the user, I for the item, R for rating, and spar for sparsity.

Table 3.1. *Datasets Detail (movielens.umn.edu)*

Name	Scale	U	I	R	Spar (%)
ML 100K	1-5 St	943	1682	100000	93.00
ML 1M	1-5 St	6040	3706	1000209	95.00
ML 20M	0.5-5 St	7120	27278	1048575	99.00
User-based model ML 20M	0.5-5 St	1500	7500	676312	93.96
Item-based model ML 20M	0.5-5 St	7000	1500	781798	92.55

3.4.2. Items' and users' weight: Bayesian posterior mean method (BPM)

We require the weights of items as outcomes in the user-based regression model and the weights of users as outputs in the item-based regression model. For those purposes, we employed the Bayesian Posterior Mean technique (BPM). BPM is a statistical theory based on Bayesian interpretation probability used to determine the degree of confidence in an occurrence. The level of trust may be determined using prior knowledge about the event or past experiments. The following formula is used to determine the weight of each item mathematically (the item weight I_w comparable to a Bayesian posterior mean) ('IMDb' 2019):

$$I_w = \left(\frac{U_i}{U_i + min(U_i)} \cdot \mu_i\right) + \left(\frac{min(U_i)}{U_i + min(U_i)} \cdot \mu\right) \quad (3.7)$$

Where:

U_i: The collection of all users who rated the item i.

$min(U_i)$: The minimal number of ratings necessary to include i in the specified dataset.

μ_i: The item's mean rating, and

μ: The average rating across a dataset.

While the following equation is used to determine the weight of each user:

$$U_w = \left(\frac{I_u}{I_u + min(I_u)} \cdot \mu_u\right) + \left(\frac{min(I_u)}{I_u + min(I_u)} \cdot \mu\right) \quad (3.8)$$

Where:

I_u: denotes the collection of items that are rated by u.

min (I_u): the minimum wanted ratings from u to be involved in the dataset;

μ_u: the average of $u's$ ratings; and

μ: the average of all ratings in the dataset.

3.4.3. Evaluation metrics

It is very crucial to predict rating in machine learning models by evaluation metrics. Three different assessment metrics are employed in this study to measure the performance of prediction: Mean Absolute Error (MAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE), as mentioned in section 2.5.

3.4.4. Experimentation methodology

The tests have been performed on Movie Lens data sets. We have created the code that controls our model using Python 3.9. It is executed on the Pycharm 2020.2.1 environment. We limited the subset of the ML-20M data set to avoid the data sparsity problem, thereby lowering the complexity of the model and the amount of running time.

The first 1500 users with the greatest rating number in the user model set are contained in this sub-set. The highest user owns 2711 ratings, and the lowest user has 190 ratings. We concentrate on the first 1500 items in the data set for the item-based model with the highest rating. The top item has a rating of 3498, and the lowest item has a rating of 165. Then, the BPM was determined for each user and item, and the average effect (slope) of each entity was derived using a multi-regression coefficient value.

The weight of items must be measured in the user-based similarity model to measure the influence of the user rating on the item weight. The weight of users is calculated for the item-based similarity model. We apply the suggested K-NN rating model on the specified sub-datasets for the rating prediction procedure. In the user-based and item-based models, K has a dynamic value. Then, we utilized the prediction rating model for 250 ratings from each user. We employed the same methods in the item-based model.

To test the proposed model quality using the selected dataset, the NNs were arranged according to:

Firstly, the PCC degree of users /items with the active user /item, where PCC value >=0.5. Secondly, the slope difference between the active user (item) and the NNs, where the difference is <=0.5. Thirdly, the difference in R-Squared (R^2) value between the active user (item) and NNs where the difference is <=0.5; (R^2 is a mathematical pointer expressing a relationship of the variance of a dependent variable specified in regression analysis by the independent variable or variables. It explains to what range the variance of one variable explains the variance of another variable (HAYES 2020)). Fourthly, the variance in R-squared (R^2) between the active user (item) and his correlated users (i.e., according to PCC and R^2) where the difference is <=0.5. Finally, the proposed model (PCC and regression) where the difference is <=0.5.

3.4.5. Results and discussion

The model's findings were studied and compared with various contemporary CF similarity metrics to understand why the neighbor selection model is based on the PCC-Slope.

First, we can see from the rating vectors of the nearest neighbors that resulting from suggested models' that the current user has the same tastes as the NNs. They have similar preferences and dislikes. The contemporary item in the item-based model has similar rating values to its closest neighbors, indicating that the user likes the active item and its similar items (its neighbors), as seen in Tables 3.2 and 3.3.

Table 3.2. *A representative NN sample for a*

a	U1	U2	U3	U4
5	5	5	5	5
5	5	4.5	0	5
5	5	4	4	5
5	4.5	4	4	5

Table 3.3. *A representative NN sample for I*

I	I1	I2	I3	I4
4	5	4	4	5
4	3	4	5	4
4	3	4	3	4
5	5	4	5	4

We hypothesize that by adding the value of significance of connected entities, we may boost the information available to CF methods, get more suitable NNs, mitigate the data sparsity, and improve RS accuracy. Thus, the first phase of our studies demonstrates an increase in accuracy when our model is used for user-based rating prediction compared to other models. We illustrate the difference in the prediction accuracy of user ratings as compared to the specified models using an average MAE value of only ten users.

The experimental results validate the accuracy of our suggested NNs selection and rating prediction approaches. As seen in Figure 3.2, our suggested model outperforms the other models since it achieves the lowest MAE value with virtually all users; MAE= 0.28 with U4, followed by the PCC+ Rsqr model. The PCC model records the worst MAE, which is 0.68 for the same person.

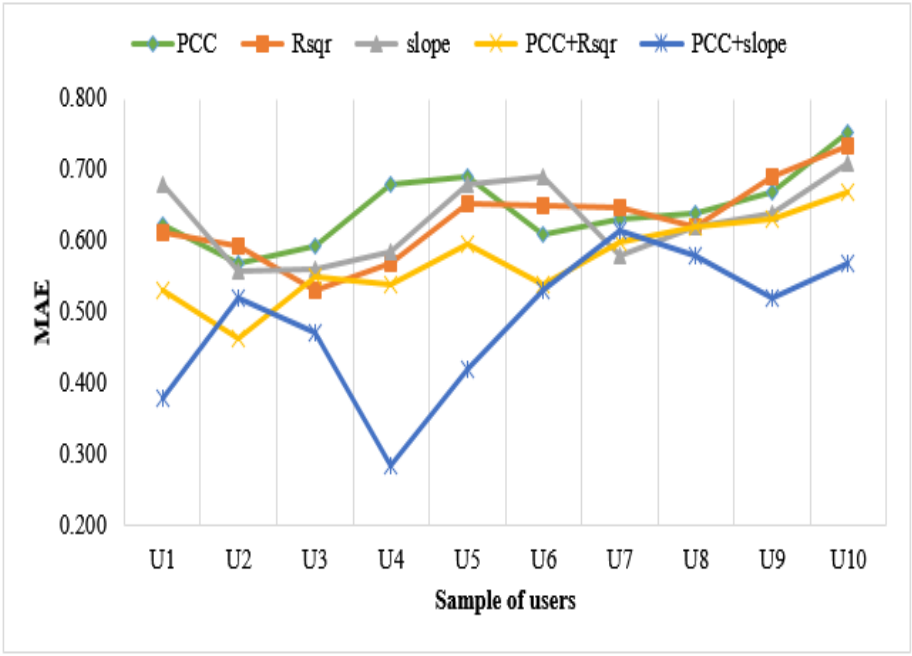

Figure 3.2. *MAE comparison between our user-based model and other models.*

As a result, we assert that relying only on the correlation model, slope value, or R-squared value is insufficient for determining the proper neighbors in the RSs. Thus, by combining the suggested model with conventional entity similarity, we enhanced the amount of information in the CF method and verified our hypotheses. In the item-based model, we achieved the same findings; our model remained to have the lowest MAE value compared to the rest of the models (Figure 3.3).

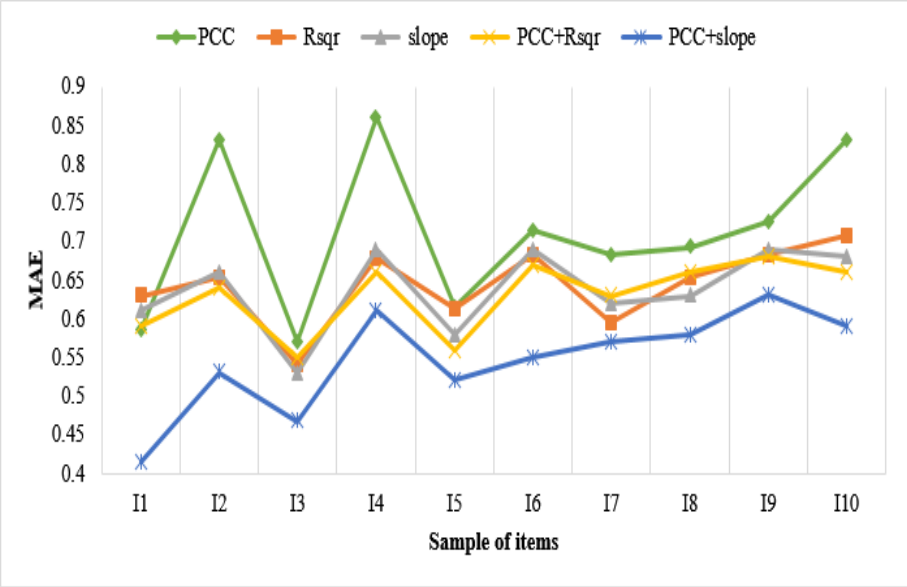

Figure 3.3. *MAE comparison between our* item-*based model and other models.*

Although we set 50 neighbors (K=50), neither the user-based nor the item-based model's rating prediction value exceeded K=5.

Tables 3.4 and 3.5 detail the assessment criteria for the user- and item-based models using the MAE, MSE, and RMSE. In both tables, our suggested models outperform the other models in terms of prediction accuracy. M1: PCC; M2: R-sqr; M3: slope; M4: PCC+Rsqr; and M5: PCC+slope.

Table 3.4. *User sample and assessment metrics*

	MAE					MSE					RMSE				
U	**M1**	**M2**	**M3**	**M4**	**M5**	**M1**	**M2**	**M3**	**M4**	**M5**	**M1**	**M2**	**M3**	**M4**	**M5**
1	0.6	0.6	0.6	0.4	0.3	0.5	0.5	0.4	0.4	0.4	0.7	0.7	0.6	0.6	0.6
2	0.6	0.6	0.6	0.5	0.4	0.6	0.6	0.5	0.6	0.4	0.7	0.8	0.7	0.8	0.6
3	0.7	0.7	0.6	0.5	0.4	0.6	0.6	0.5	0.6	0.5	0.8	0.8	0.7	0.7	0.7
4	0.6	0.6	0.6	0.5	0.4	0.7	0.6	0.6	0.6	0.5	0.8	0.8	0.7	0.8	0.7
5	0.6	0.5	0.6	0.6	0.5	0.7	0.5	0.4	0.5	0.4	0.8	0.7	0.6	0.7	0.7

Table 3.5. *Sample of items and their evaluation metrics*

MAE	**MSE**	**RMSE**

I	M1	M2	M3	M4	M5	M1	M2	M3	M4	M5	M1	M2	M3	M4	M5
1	0.6	0.6	0.6	0.6	0.4	0.6	0.7	0.7	0.7	0.5	0.8	0.8	0.8	0.7	0.7
2	0.8	0.7	0.7	0.6	0.4	0.9	0.7	0.7	0.7	0.5	0.9	0.8	0.8	0.7	0.7
3	0.6	0.5	0.5	0.6	0.5	0.6	0.6	0.6	0.6	0.5	0.8	0.8	0.8	0.7	0.7
4	0.9	0.7	0.7	0.7	0.5	0.9	0.7	0.8	0.9	0.6	1.0	0.8	0.9	0.7	0.7
5	0.6	0.6	0.6	0.6	0.5	0.7	0.7	0.7	0.7	0.6	0.8	0.8	0.8	0.8	0.7

Second, we contrast the user-based and item-based models. According to the assessment metrics values, the accuracy of the user-based model surpasses the accuracy of the item-based model, as the user-based model records MAE=0.497. In contrast, the item-based model registers MAE=0.52 (Figure 3.4).

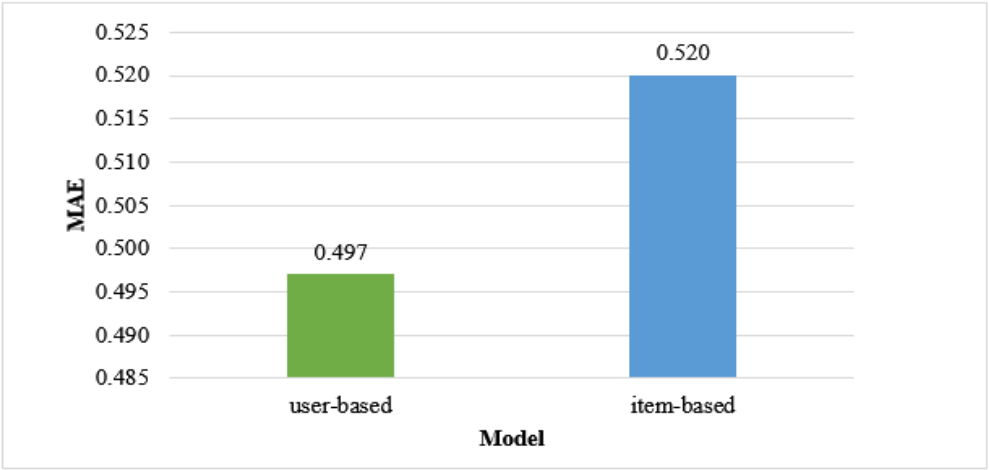

Figure 3.4. *MAE averages for user and item models*

Figures 3.5 and 3.6 depict these comparative results in terms of MSE and RMSE. As a result, we've decided to design an accurate recommendation system around users rather than items.

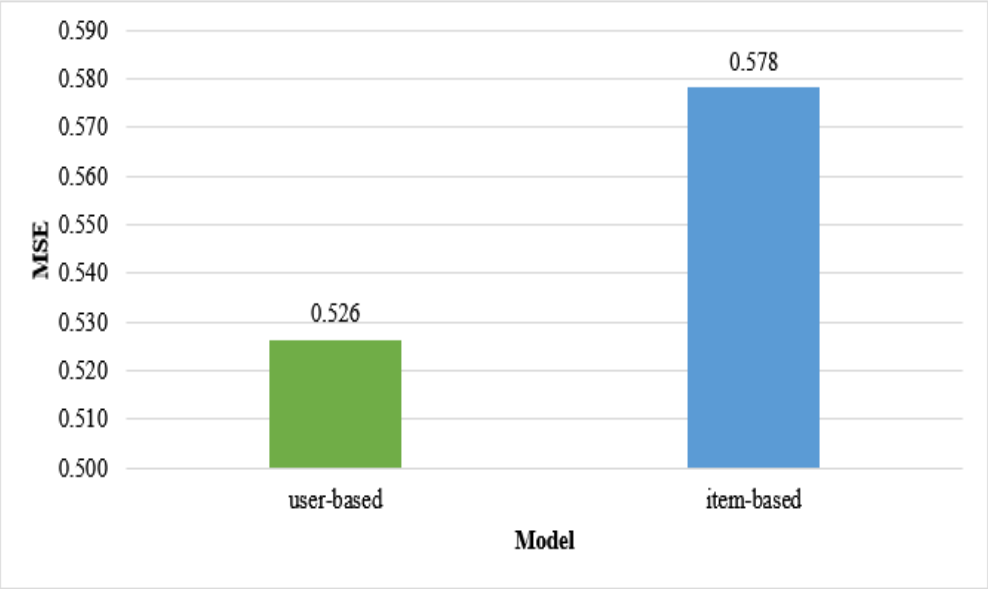

Figure 3.5. *MSE averages for user and item models*

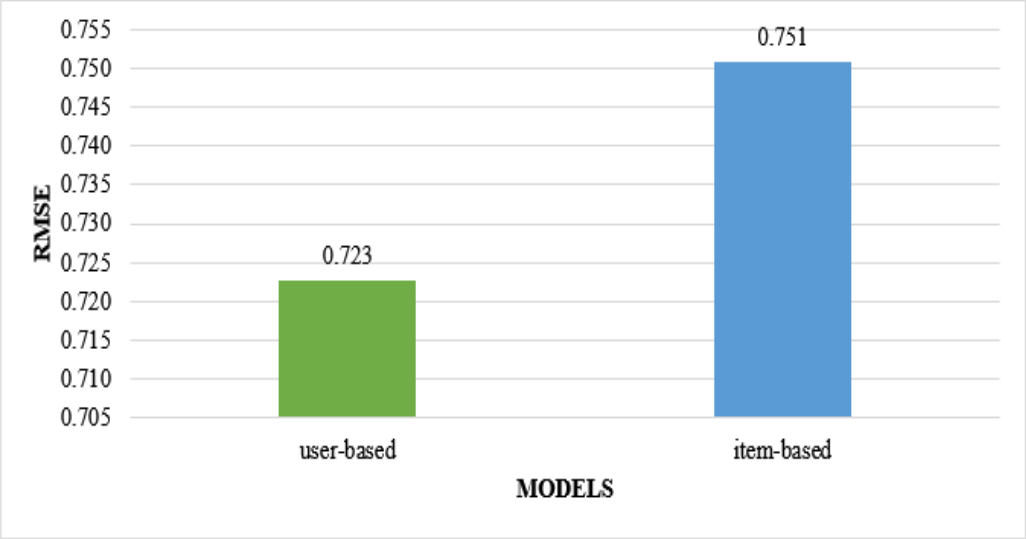

Figure 3.6. *RMSE averages for user and item models*

As previously stated, several types of research have been conducted to increase the efficiency of CF systems, whether by increasing the process of neighbor collecting or by using current assessments of similarity. To demonstrate that the proposed technique improves the accuracy of the CF algorithms' suggestions, we compared the proposed method to earlier studies.

Wang et al. (2020) combined structural and rating-based similarity estimates in their model (D. Wang, Yih, and Ventresca 2020). We compared our suggested models'

precision and effectiveness to their model using the same number of movie lens data sets. Tables (3.6 and 3.7) and figures (3.7-3.10) illustrate the findings.

Table 3.6. *Comparison of the predictive accuracy of proposed models to that of contemporary CF algorithms on the ML-100 K data set. MAE values are listed in descending order; our models are highlighted in bold*

Method	**Parameterization**	**MAE**	**RMSE**
ItemKNNPearson	K=60	0.736125	0.936507
ItemKNNAdjustedCosine	K=60	0.733618	0.935528
ItemKNNPearsonReg	K=60	0.733399	0.932648
ItemKNNAdjustedCosineReg	K=60	0.729538	0.929991
ItemKNNBCosine	K=25	0.725362	0.924898
BiasedMatrixFactorization	num_factors = 40, bias_reg = 0.1, reg_u = 1.0, reg_i = 1.2, learn_rate = 0.07, num_iter = 100,	0.721715	0.912668
ItemKNNPearsonShrink	K=40	0.718144	0.915559
ItemKNNAdjustedCosineShrink	K=40	0.71739	0.91571
SigmoidUserAsymmetricFactorModel	num_factors = 5, regularization = 0.003, bias_reg = 0.01, learn_rate = 0.006, bias_learn_rate = 0.7,	0.71655	0.910353
SVDPlusPlus	num_factors = 50, regularization = 1, bias_reg = 0.005, learn_rate = 0.01, bias_learn_rate = 0.07,	0.715941	0.909214
ItemKNNCombined	K=25	0.708815	0.909131
ItemKNNCombinedReg	K=25	0.704588	0.903682
PCC-slope Item-based	**Dynamic K from 1 to 50**	**0.596141**	**0.7359**
PCC-slope User-based	**Dynamic K from 1 to 50**	**0.53**	**0.65**

Table 3.7. *Comparison of the predictive accuracy of proposed models to that of contemporary CF algorithms on the ML-1 M dataset. MAE values are listed in descending order; our models are highlighted in bold.*

Method	Parameterization	MAE	RMSE
ItemKNNAdjustedCosine	K = 80	0.690876	0.87984
ItemKNNAdjustedCosine	K = 80	0.690622	0.879012
BiasedMatrixFactorization	num_factors = 120, bias_reg = 0.001, regularization = 0.055, learn_rate = 0.07, num_iter = 100,	0.675454	0.853102
SVDPlusPlus	num_factors = 20, num_iter = 80, reg = 0.05, learn_rate = 0.005	0.667725	0.853002
ItemKNNCombinedReg	K = 20	0.664384	0.852439
ItemKNNCombined	K = 20	0.66427	0.852514
PCC-slope Item-based	**Dynamic K from 1 to 50**	**0.576141**	**0.7267**
PCC-slope User-based	**Dynamic K from 1 to 50**	**0.526**	**0.68**

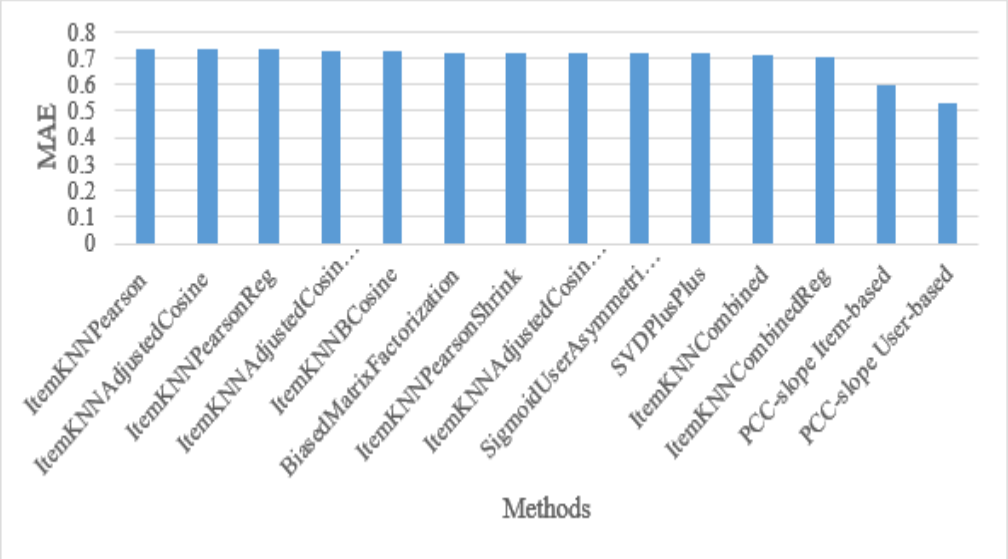

Figure 3.7. *MAE comparison of suggested models to contemporary CF techniques on the ML-100 K dataset*

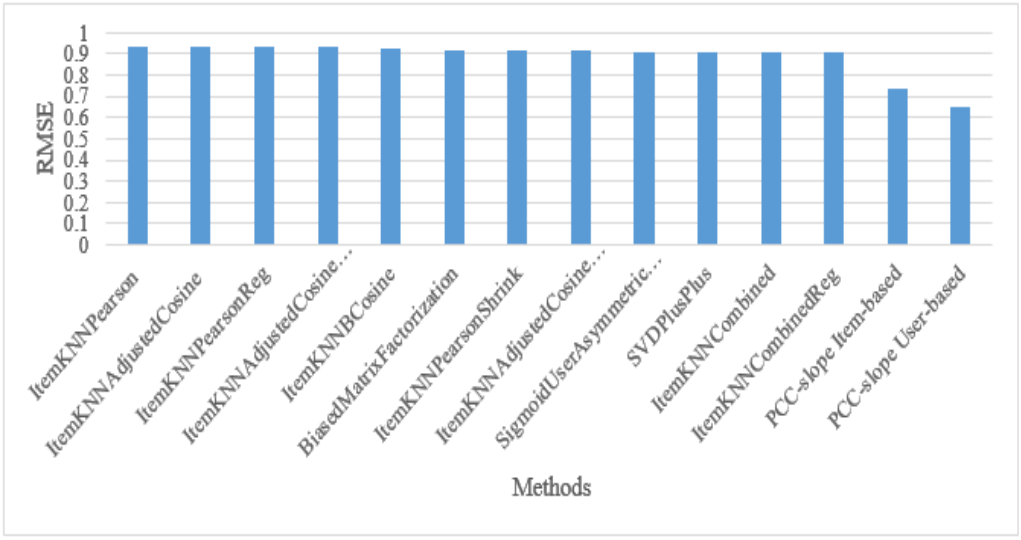

Figure 3.8. *RMSE comparison of suggested models to contemporary CF techniques on the ML-100 K dataset*

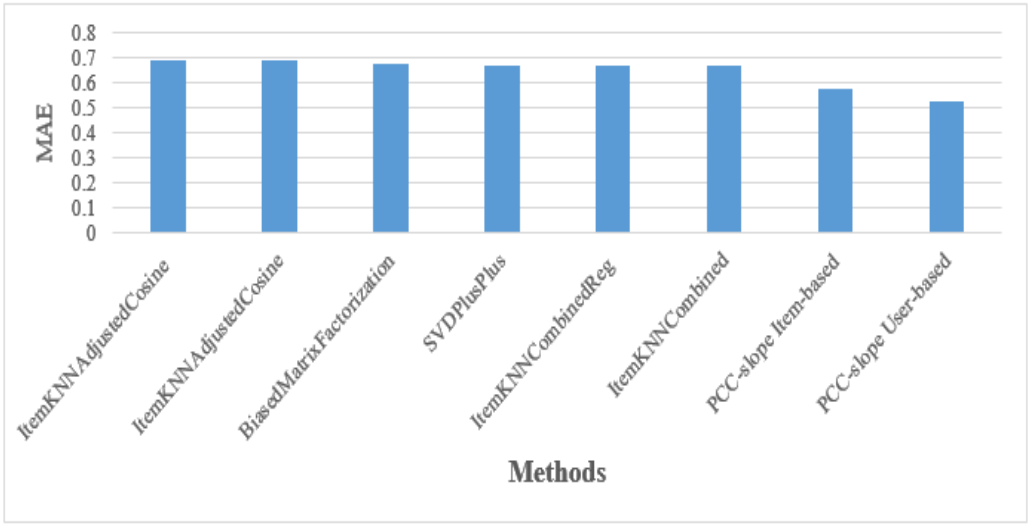

Figure 3.9. *MAE comparison of suggested models to contemporary CF techniques on the ML-1M dataset*

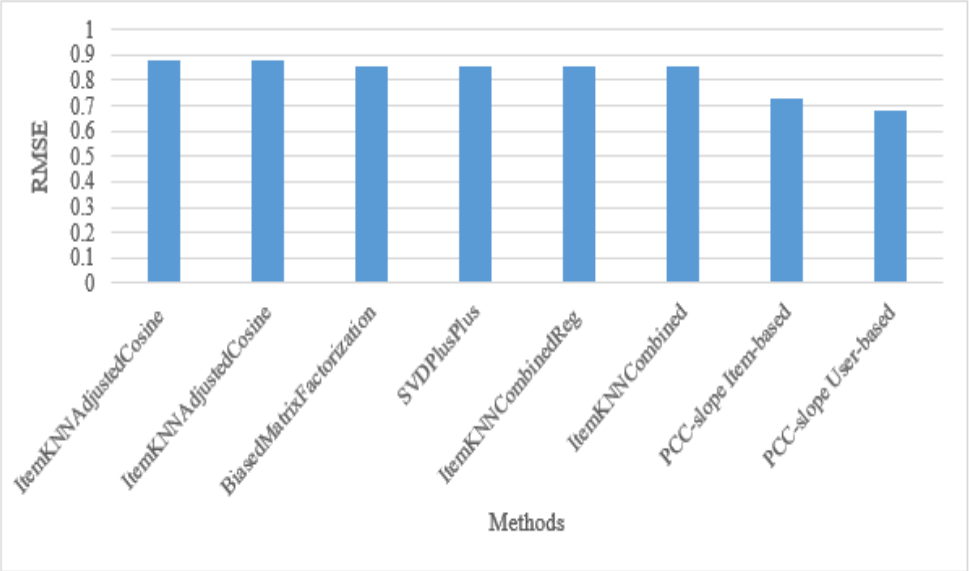

Figure 3.10. *RMSE comparison of suggested models to contemporary CF techniques on the ML-1M dataset*

As seen in the comparative tables and figures, our proposed models had reduced MAE and RMSE values, with ML-100K having an MAE of 0.53 and an RMSE of 0.65. MAE=0.52, RMSE=0.68 when using ML-1M.

3.5. Summary

RS is used to collect data on websites linked with the user's interests in various products. It utilizes multiple online information sources to ascertain consumers' preferences for certain goods; as such, it plays a critical role in the purchase of commodities or services (Xiao et al. 2018).

The experimental findings indicate that the proposed model (PCC+ Slope model) outperformed the other considered models. This finding emphasizes the critical nature of including information about recommendation system entities to locate acceptable neighbors and provide correct suggestions.

As a result, we can conclude that the suggested model is selecting nearest neighbors effectively and supporting the recommendation process correctly. Because our rating prediction model requires a limited number of NN to calculate entity rating prediction, the model's complexity and computational time are reduced.

4. A SIMILARITY MEASURE MODEL BASED ON THE DISSIMILARITY DEGREE BETWEEN USERS

Collaborative filtering is a frequently used technique for producing user-specific services. This technique's essential component is recognizing comparable users or items based on their correlations for the system to provide suggestions to users. The majority of recommendation approaches used in conjunction with collaborative filtering algorithms are based on conventional similarity metrics such as Pearson correlation coefficient and cosine. While these approaches can produce accurate forecasts, they have significant disadvantages, including scalability, locating neighbors, and dealing with the cold-starting problem. This chapter introduced a user similarity metric that improves suggestion accuracy by assessing how similar each user is even when just limited ratings are available.

The proposed model considers just users' common ratings, as determined by the fraction of different ratings and their maximum value. We conduct experiments on real data sets and compare the results to numerous state-of-the-art similarity metrics. The results demonstrate the suggested similarity model's superiority in neighbor matching, performance accuracy, and resolving user cold-starting situations. Thus, this work contributes to current research on the user cold-starting problem and assists in forming appropriate neighbors in the user-based collaborative filtering technique based on the suggested similarity measure.

4.1. Introduction

To build a good neighborhood, CF employs a range of similarities among users or items. The fundamental regard of neighborhood-based CF algorithms is to discover the similarity between users or items. The nearest neighbors (NN) are those users who have the same interests. The CF algorithm relies only on the NNs preferences to predict the target user ratings for a specific item instead of depending on the entire users' rating. Therefore, for obtaining accurate prediction accuracy, it is imperative to choose the NN carefully.

4.1.1. Problem Definition

Generally, K-NN-based CF algorithms set the neighborhood of an entity by employing traditional similarities, like PCC, COS (X. Su and Khoshgoftaar 2009), and ACOS measure (B. Sarwar et al. 2001). These similarity measures are work well at

capturing entity relationships. Nevertheless, they have shortcomings in working with a sparse dataset (Zeng et al. 2004), cold start (Middleton, Alani, and Roure 2002) problem, and can provide undependable neighborhoods. Therefore, researchers recommend various studies to enhance the CF algorithms' NN selection ability, besides improving the correctness of predictions either by offering new similarity measures (Sun et al. 2012; Jesús Bobadilla, Ortega, and Hernando 2012) or developing new prospects for NN configuration (Bellogín and Pablo Castells 2009; Koren 2010).

4.1.2. Contributions

This chapter concentrates on prediction accuracy in memory-based CF algorithms, the most extensively used algorithm in recommender systems (Gediminas, Manouselis, and YoungOk Kwon 2011). The focus of CF is to measure similarities or relationships among entities, i.e., users or items. Traditional similarity measures are not sufficient to obtain similar users correctly, particularly for the cold start problem in C.F. Hence, it is challenging to introduce a better accurate similarity measure CF-based RSs.

Although we have the same goal as previous researchers in finding a similarity measure between users for the recommendation system, our research is different. The proposed model determines the degree of difference or dissimilarities between users' ratings in finding similarities. The dissimilarity degree is obtained by determining the weight of the dissimilar ratings in the co-rated items and determining the most considerable difference in the co-rated rating between users. To our knowledge, these factors have not been addressed in the previous researches.

Additionally, the proposed model requires nothing more than user ratings that CF systems essentially use. In contrast to most previous research that needs multiple information to construct the similarity model, such as user context information, user registration information, user general behavior information, and others. It strives to promote recommendation performance when there are a small number of ratings in the user record by forming a new similarity measure for CF systems based on the weight of dissimilarity in the user rating vector. Thus, this study can be considered as complementing existing researches on cold-start problems and forming suitable neighbors in the user-based CF method based on this similarity measure.

The main contributions of this research to the recommender system are the following points:

1- Determining the degree of rating difference between pairs of users and the weight of this difference according to the data set's highest rating value.

2- Building a new similarity measure model that can select the nearest neighbors accurately and reliably by investing in the rating differences.

3- Enhancing recommendation performance in cold-starting situations, where only a few ratings are available for similarity computation for each user.

4- Experimental analysis on two real-world data sets revealed that the newly suggested similarity measure model surpasses the other traditional similarity measures in prediction accuracy and recommendation quality by picking the appropriate neighbors.

4.2. A New Similarity Measure Model

We begin in this part by examining the limitations of existing similarity metrics. Following that, we will discuss the motivations for and interpretation of the intended similarity measure technique. Finally, we demonstrate the mathematical formalization of the innovative similarity measure approach that we have proposed. We suppose that $U = \{u1, u2 \dots uN\}$ and $P = \{p1, p2 \dots pM\}$ are referred to the collection of users and items, respectively. The symbol R denotes the user-item rating matrix as $R = (r_{i,j})_{NxM}$ I=*1, 2…, N; j=1, 2.., M.*

4.2.1. The insufficiency of present similarity measures

The traditional similarity measures have some drawbacks. Researchers have proposed many modifications to these scales to overcome these defects, including the Constrained Pearson Correlation Coefficient (CPCC) (D. E. Chen 2009) to measure the effect of positive and negative ratings.

If two people share a large number of goods, their similarities will be more realistic. The weighted Pearson correlation coefficient (WPCC) and the sigmoid function based on the Pearson correlation coefficient (SPCC) are provided for this purpose (J. L. Herlocker et al. 1999). The WPCC and SPCC formulae are as shown below:

$$sim(u,v)^{WPCC} = \begin{cases} sim(u,v)^{PCC} \cdot \frac{|I|}{H}, & |I| \leqslant H \\ sim(u,v)^{PCC}, & \text{otherwise} \end{cases} \quad (4.1)$$

$$sim(u,v)^{SPCC} = sim(u,v)^{PCC} \cdot \frac{1}{1 + exp\left(-\frac{|I|}{2}\right)} \quad (4.2)$$

Where: H is an experimental value (J. L. Herlocker et al. 1999).

Each user has a special rating style, some of them do not give a high rating even if he prefers the item, and others provide a high rating even if he is not interested in the item. This style does not account for the traditional cosine similarity (COS) method. Therefore, the adjusted cosine measure (ACOS) (B. Sarwar et al. 2001) has been proposed to process those drawbacks. The ACOS is described in Eq. (2.3).

Jaccard measure (Koutrika 2009) also is commonly used in the similarity measure. According to the notion that people are more identical if they share a greater number of ratings, this metric solely considers the number of shared ratings between users. It does not, however, take into consideration the importance of ratings. But on the other side, the mean squared difference (MSD) similarity metric (Cacheda et al. 2011) is concerned with the value of the ratings, but it disregards the credibility of common ratings. The formulas are as follows:

$$sim(u,v)^{\text{Jaccard}} = \frac{|I_u| \cap |I_v|}{|I_u| \cup |I_v|} \quad (4.3)$$

$$sim(u,v)^{MSD} = 1 - \frac{\sum_{p \in I} \left(r_{u,p} - r_{v,p}\right)^2}{|I|} \quad (4.4)$$

Where:

I_u, and I_v: denotes the set of items rated by user *u* and *v,* respectively.

While new similarity criteria have been proposed and overcome some of the problems of traditional similarity approaches, these similarity approaches have significant drawbacks too.

This section will describe various similarity approaches and point out their shortcomings. A user-item rating matrix is shown in Table 4.1.

Table 4.1. *A sample of the user-item rating matrix*

	I1	I2	I3	I4
U1	4	3	5	4
U2	5	3	-	-
U3	4	3	3	4
U4	2	1	-	-
U5	4	2	-	-

Within the practices, we assume four items (I) and five users (U). The notation represents the rating matrix's missing ratings "-"; then, similarly with the similarity metrics described previously, we calculate the similarities of users inside Table 4.1. The results of the user similarity analysis in Table 4.1 are depicted in Table 4.2.

Table 4.2. *The user similarity of Table 4.1 according to similarity measures*

	U2	U3	U4	U5
U1	0.71	0.0	0.71	0.71
U2		1.0	1.0	1.0
U3			1.0	1.0
U4				1.0

(a) PCC

	U2	U3	U4	U5
U1	1.0	0.58	-0.45	0.71
U2		1.0	-0.45	0.71
U3			-0.45	0.71
U4				0.32

(b) CPCC

	U2	U3	U4	U5
U1	0.52	0.0	0.52	0.52
U2		0.73	0.73	0.73
U3			0.73	0.73
U4				0.73

(c) SPCC

	U2	U3	U4	U5
U1	0.61	0.98	0.61	0.61
U2		0.70	0.99	0.99
U3			0.69	0.69
U4				1.0

(d) COS

	U2	U3	U4	U5
U1	-0.363	0.0	-0.162	-0.316
U2		0.171	0.077	0.77
U3			0.224	0.224
U4				0.1

(e) ACOS

	U2	U3	U4	U5
U1	0.98	0.96	0.84	0.98
U2		0.98	0.74	0.96
U3			0.84	0.98
U4				0.9

(f) MSD

	U2	U3	U4	U5
U1	0.5	1.0	0.5	0.5
U2		0.5	1.0	1.0
U3			0.5	0.5
U4				1.0

(g) Jaccard

Due to the symmetry of the user similarity matrix, we can only provide partial interpretations. The following are the primary disadvantages:

1- Despite identical ratings by two people, there is little similarity.

According to PCC, Table 4.2(a) depicts the user correlation matrix. As seen in Table 4.1, the U1 and U3 have fairly comparable ratings. For U1 and U3, the rating vector is (4, 3, 5, 4) and (4, 3, 3, 4), respectively. However, as seen in Table 4.2(a), the similarity between these two users is zero. This disadvantage is still in the SPCC and the ACOS as well (Table 4.2 (c) and (e)). The CPCC has marked a bit rise as the similarity is 0.577 (Table 4.2 (b)).

2- Disregarding the common rating proportion between users drives low similarity accuracy.

MSD calculates the average disagreement between two users but ignores the fraction of people that agree on a rating. As a result, this may account for the low accuracy. The similarity between U1 and U2 is equal to 0.98, whereas U1 and U3 are equal to 0.96. We can see that U1 and U3 should be more than or similar to the similarity between U1 and U2, as the MSD does not consider the user's common rating proportion.

In Table 4.1, the common rating proportion between U1 and U2 is 1/3, while the ratio is 1/2 between U1 and U3. The fraction of shared ratings is computed as the similar ratings divided by the number of ratings for both users.

3- It is difficult to discover users' similarity degree when disregarding the value of a common rating and depend only on the number of the common rating.

The Jaccard similarity studies the number of common ratings and does not examine a common rating's value; this drives the inability to discover the users' similarity degree. In Table 4.2 (g), we perceive that similarities: either 1.0 or 0.5. The similarity between U1 and U2 is 0.5; because U1 rated four items, and U2 rated two items. While the similarity between U2 and U5 is 1.0. Additionally, the similarity between U2 and U4 is 1.0 too, although the latter users should be lower similarity than the former users.

4.2.2. The motivations the proposed neighbor selection model

The drawbacks of conventional similarity metrics and processed variations are discussed in Section 4.2.1. Many recommender systems allow users to evaluate a small number of items. The insufficient amount of preferences results in incredibly low accurate

likeness established in previous models. The proposed research proposes an enhanced similarity measure model to increase the effectiveness of recommender systems.

The motivations for the new neighbor selection model are as follows:

- The similarity measure model examines the common dissimilar rating value, the weight of the common different ratings, and the largest difference in the common different ratings.

- Normalization and efficient merging of the standard association model with additional association measure models are required.

4.2.3. Rating dissimilarity and largest difference similarity measure

We give the mathematical formalization of the proposed novel similarity measure technique in this part. To penalize incorrect similarity and reward correct likeness, we employ a linear function in our model. Moreover, we will call the new models RDLD (Rating Dissimilarity and largest Difference). The RDLD similarity model can be calculated as follows:

$$sim(u,v)^{RDLD} = 1 - \left(w_{RD}(u,v) * w_{LD}(u,v)\right) \tag{4.5}$$

Where: $w_{RD}(u,v)$ represents the dissimilarity weight of common rating between *u* and user *v*. it is calculated as the Eq. (4.6):

$$w_{RD\ p\in I}\left(u_p, v_p\right) = \frac{\#\ of\ dissimilar\ rating_{p\in I}(u_p,v_p)}{\#\ of\ I} \tag{4.6}$$

Where:

$\#\ of\ dissimilar\ rating_{p\in I}(u_p, v_p)$: Means the number of the rating of *u* on item *p* that is not equal to the rating of user v on item *p*; i.e. $r_{u,p} \neq r_{v,p}$; and

$\#\ of\ I$: represents the number of common items or ratings.

The second part in the equation (4.5) is $w_{LD}(u,v)$ which represents the weight of the largest difference between common dissimilar ratings. It is calculated as:

$$w_{LD\ p\in I}\ (u,v) = \frac{Max\ (|differece_{p\in I}(u_p,v_p)|)}{Max\ rating\ in\ the\ data\ set} \tag{4.7}$$

Where: $Max\ (|differece_{p\in I}(u_p, v_p)|)$ means the maximum value of the absolute difference or subtraction operation between dissimilar rating $Max(|r_{u,p} - r_{v,p}|)$; and

$Max\ rating\ in\ the\ data\ set$: represents the max rating in the dataset. For example, the movie lens dataset's max rating is 5; in Yahoo! Music dataset is 100.

It is clear that the RDLD model is composed of two factors; the weight of unequal or dissimilar ratings in common ratings and the importance of the largest difference in those unequal ratings. Each part belongs to (0, 1) in this model. These two factors composed the degree of difference or dissimilarity between users; thus, we can calculate the similarity of users by subtracting their differences from one, as illustrated in Eq. (4.5). The RDLD similarity value is between 0–1; the higher value indicates higher similarity. Algorithm 4.1 displays the operations of the RDLD model and prediction methodology.

Algorithm 4.1: Recommendation procedure

Input: User rating

Output: Rating prediction $P_{u,p}$

1- Divide the dataset into 80% from users as training users and 20% as testing users.
2- Select a user from the testing set as an active user *u* and their rating on *p*.
3- Determine the common items between users in the training set.
4- Count the number of dissimilar ratings in the common item ratings.
5- Calculate the dissimilarity weight in the common rating by Eq. (4.6) to find $(w_{RD\ p\in I}\ (u_p, v_p))$.
6- Calculate the largest difference between dissimilar ratings by subtracting the dissimilar common rating values between users; $Max(r_{u,p} - r_{v,p})$;
7- Calculate the weight of step (6) by Eq. (4.7) to find $w_{LD\ p\in I}\ (u, v)$.
8- Multiply the result in step (5) by the result in step (7) to obtain the dissimilarity degree between users.
9- Apply the Eq. (4.5) to obtain the similarity value between users, i.e. $sim(u, v)^{RDLD}$.
10- Sort the users' similarities in descending order.
11- Calculate the rating prediction of u for p by applying the rating prediction model (J. K. S. Al-safi and Kaleli 2021) in section 3.3.2.
12- Return the rating prediction value of *u* on *p*; $P_{u,p}$
END Procedure

In Section 4.2.1, we discussed the drawbacks of existing similarity metrics. We demonstrate that the new similarity model (RDLD) can overcome these drawbacks, as in Table 4.3.

Table 4.3. *The user similarity of Table 4.1 according to the RDLD similarity measure*

	U2	**U3**	**U4**	**U5**
U1	0.9	0.9	0.6	0.9
U2		0.9	0.4	0.8
U3			0.6	0.9
U4				0.6

First, we can see from Table 4.3 that U1 and U3 are identical to U1 and U2 because they both have the same amount of different ratings. This difference is not accounted for in the PCC, CPCC, SPCC, ACOS, and MSD similarity measures, implying that the RDLD similarity measure can overcome poor similarity despite comparable evaluations by two users.

Second, U3 and U5 are more comparable than U4 and U5; this is not discovered using COS, ACOS, Jaccard, or JMSD. Thus, this demonstrates that the RDLD similarity measure may avoid this error.

Third, if there is a significant disparity in two users' ratings, the similarity will be rather small. For instance, the similarity between U2 and U4 is rather low; it is around 0.4. It is the user similarity matrix's lowest similarity. Indeed, as seen in Table 4.1, they share the least resemblance. However, the RDLD model takes note of this small resemblance as well.

4.3. Experiments

The tests were conducted using the Movie Lens and Yahoo! Music datasets to determine the effectiveness of the proposed NN selection model in the RDLD. The trials assessed the accuracy of RDLD for users and analyzed user cold-starting scenarios. Additionally, the RDLD model is compared to existing CF methods.

The performance evaluation of the given model has been done on the following data sets: Movie Lens (ML-20M) and Yahoo! Music. The information of the movie lens datasets is described in section 3.4.1; while information of the Yahoo! Music dataset is shown in Table 4.4. (The int: integer)

Table 4.4. *Data sets Detail (http://research.yahoo.com)*

Name	S	U	I	R	Spar (%)
Yahoo! Music	1-100 int	1,948,882	98,211	11,557,943	99.993
User-based model Yahoo! Music	1-100 int	1,500	13,000	400,000	97.94

The ML-20M data set comprises the first 1500 users in the data set with the highest ratings. The highest-rated individual has 2711 ratings; the lowest-rated individual has 190 ratings. While the Yahoo! Music dataset also includes the first 1500 users, the highest-rated user has 4,966 ratings, while the lowest-rated person has 108 ratings.

Performance measures control the suggested method in real conditions or get on for real specific data. In our experiment, we used prediction performance measures. Numerous measures are utilized to assess system performance. They often choose to estimate the system's accuracy by determining the accuracy of computed predictions related to genuine user ratings. The popular way to assess accuracy is based on a statistical metric, like MAE and RMSE, to estimate the prediction performance as described in section 2.5.

4.3.1. Experiment methodology

The ML 20M dataset was randomly divided into 20% of users as a testing set and the remaining as a training set in all experiments. Then, we randomly selected 150 ratings from each test user and replaced their value with 0. We measured prediction accuracy using a prediction model proposed in (J. K. S. Al-safi and Kaleli 2021), with K ranging from 1 to 50.

4.3.2. Results and discussion

Many performances were shown to demonstrate the effectiveness of the RDLD similarity measure concentrating on examining recommendation performance on selecting proper neighbors and new user cold-starting situations. In the first experiment, the RDLD is compared to the PCC and COS measures in terms of the largest difference (LD) value and MAE. The RDLD is then compared against a variety of memory- and model-based methods. The second experiment simulates artificial cold starts and evaluates the results of the metric mentioned above. Among the most important steps that affect the CF algorithms' performance are selecting appropriate neighbors and predicting

the user ratings correctly. We will evaluate the performance of our model according to these two steps.

1- Experiments over full selected ratings

We applied a prediction model that relies on a dynamic value for K to improve prediction accuracy and increase recommendations' correctness.

- **Comparison analysis of the largest difference (LD) value between ratings:**

Intending to illustrate how the RDLD model supports improving NN size and quality, we presented the NN analysis on RDLD and the traditional similarity measures like PCC, and COS, according to the LD value in common rating value between users. Firstly, we calculated the LD value in common rating between users on the ML-20M data set. Secondly, we applied RDLD on this dataset to measure the similarity between users, and we sorted this similarity in descending order. Thirdly, we summarized the first 50-NN for the active user and recorded their LD value according to the sorted similarity value with other users. Then, we recurred the same steps for PCC and COS similarity measures. We demonstrate this for one user to facilitate the operation explanation. The outcomes are represented in Figure 4.1.

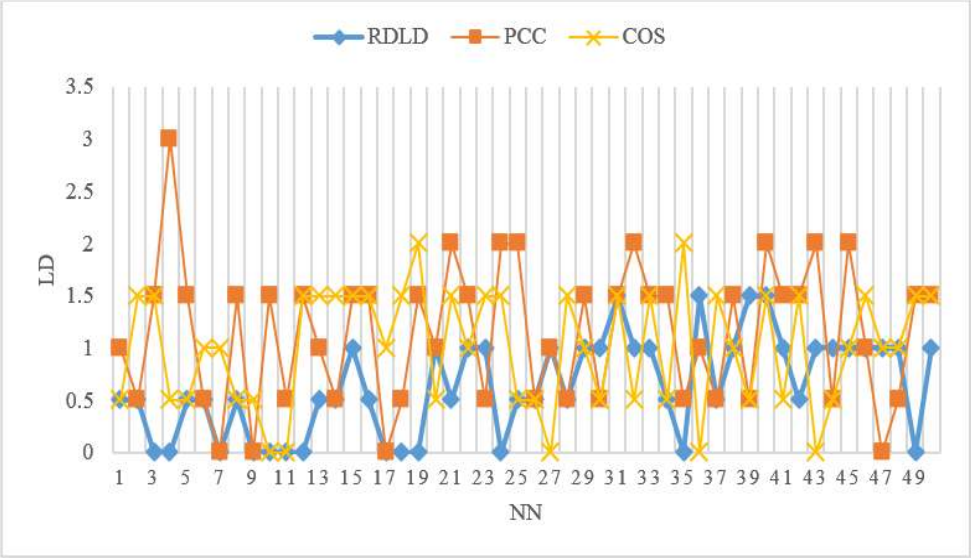

Figure 4.1. *Largest difference (LD) with NN on ML-20M dataset*

In figure 4.1, we can note that the RDLD model chose appropriate and closer neighbors that are not different from the active user in the ratings; the LD value falls in the interval (0, 1) for most of the closest neighbors selected at the head of the NN list.

The LD exceeded one in 4 of 50 chosen neighbors at the NN list's tail in the neighbors who follow the 30th-ranked neighbor.

On the other hand, the COS model selected NN who have LD=0.5 to 1.5. However, as average, the RDLD model scored LD= 0.85/5 between the active user and their closest neighbors. COS follows it with LD=1.09/5 and LD=1.23/5 for PCC models. Thus, RDLD overcomes the traditional models in selecting suitable NN. However, The PCC model is the worst model for selecting NN. Consequently, Figure 4.2 shows the sequence of neighbors for a specific user with the LD's value in the common ratings on the Yahoo! music data set.

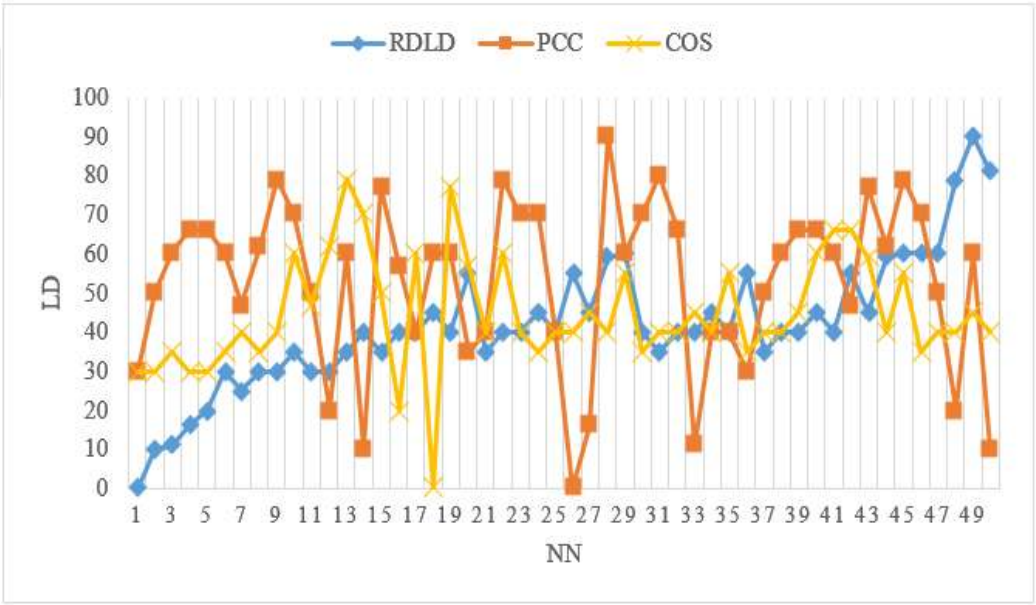

Figure 4.2. *Largest difference (LD) with NN on Yahoo! music dataset*

From Figure 4.2, we can perceive that the model RDLD succeeded in selecting the correct and appropriate neighbors regularly, as the selected first neighbor has LD= 0. The selected second neighbor has only 10 out of 100 (100 is the largest rating value in the music dataset), and the LD continued to increase regularly until the last neighbors, i.e., the fiftieth neighbor, where the last neighbor scored 81 out of 100 as the largest LD with the same specific user. While the models PCC and COS are troubled in selecting the right neighbors. According to the PCC similarity measures, the target user's first neighbor is different from the target user by LD=30 and the second neighbor by LD=50. The differences in LD values were irregular until the last neighbor. We can notice the same problem with the COS model, where the first neighbor recorded LD=30. The neighbor's

selection process is continued with unstable differences in LD value, as the selected neighbor who has the value of LD =0 is selected at the 26^{th} and 18^{th} position in PCC, and COS respectively. Nevertheless, with the RDLD model, the first selected neighbor has the value of the LD =0 with the target user.

Therefore, as average, the RDLD model scored LD=48/100 between the active user and their closest neighbors. It is followed by the COS model with LD=54.86/100, then and LD=58.76 for PCC models, respectively. However, compared with the ML-20 M data set, from 5-stars ratings, the Yahoo! music data set's results will be as: RDLD model scored LD=2.4/5, COS with LD=2.743/5, finally the PCC with LD=2.938/5. The LD. values with the ML-20 M dataset are less than in Yahoo! music dataset. Thus, the prediction accuracy with the ML-20M dataset might be better than with Yahoo! music dataset. Consequently, the results demonstrate the ability of the RDLD model to select the right neighborhood.

- **Comparison analysis of MAE:**

We want to improve the prediction accuracy of RSs. We checked the RDLD model's achievements and the traditional similarity models according to the absolute error (AE) in rating prediction. Firstly, in the experiments, we measured the similarity between users according to the RDLD, PCC, and COS to select NNs. Then, we calculated the rating prediction of these models according to the prediction mentioned above model. Finally, we summarized the AE for each rating prediction. We show this procedure for a user to describe the guesses of models clearly. We introduced the results of the ML-20M data set for the models mentioned above in Figure 4.3.

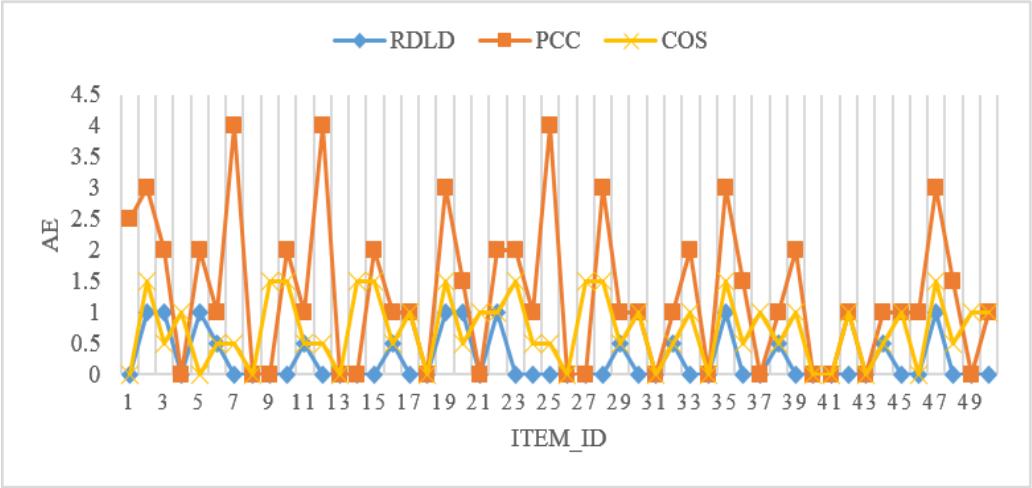

Figure 4.3. *AE for a user in ML-20M dataset*

From the results, we note that the RDLD similarity model has remarkable improvement. It obtains the lowest AE value than the other similarity measures. The AE value in the RDLD model fall between 0 to 1 values, followed by the COS model, which is fall 0 to 2 as AE While, in the PCC model, the AE value falls between 0 to 4. On average, the MAE and RMSE of the tested users are 0.493 and 0.6733 respectively by the RDLD model, which is the best. At the same time, MAE is 0.63 by COS similarity measure. However, PCC recorded 0.696 and 0.8355 as MAE and RMSE values, respectively; it is the worst model, as illustrated in Figures 4.4 and 4.5. That is because similarity measure models are sensitive to the unsuitable neighbors, where the similarity value between two users is high, but their favorites are not alike in reality.

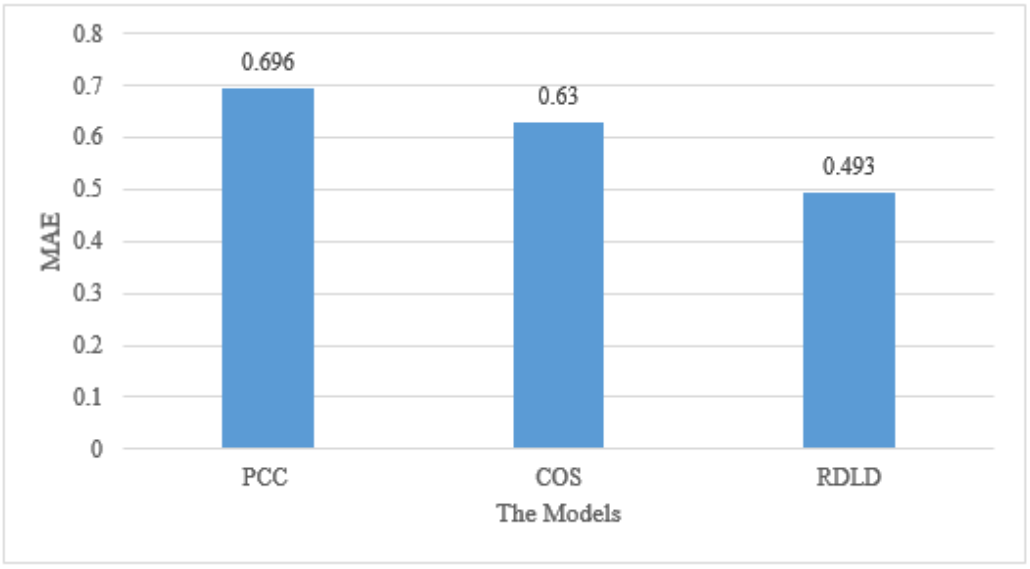

Figure 4.4. *MAE of models on ML-20M dataset*

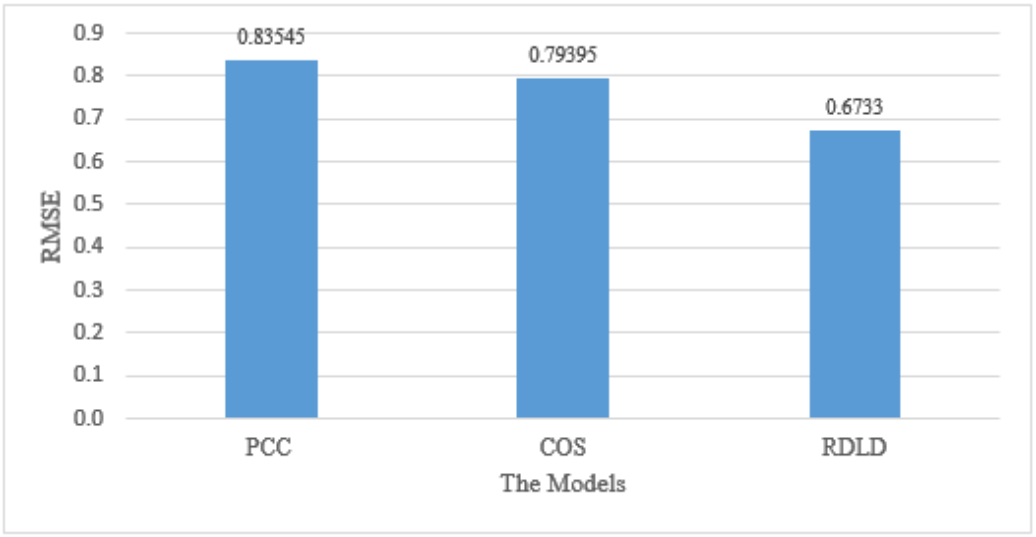

Figure 4.5. *RMSE of models on ML-20M dataset*

Figure 4.6 shows the AE value of a user on 50 items on Yahoo! music dataset.

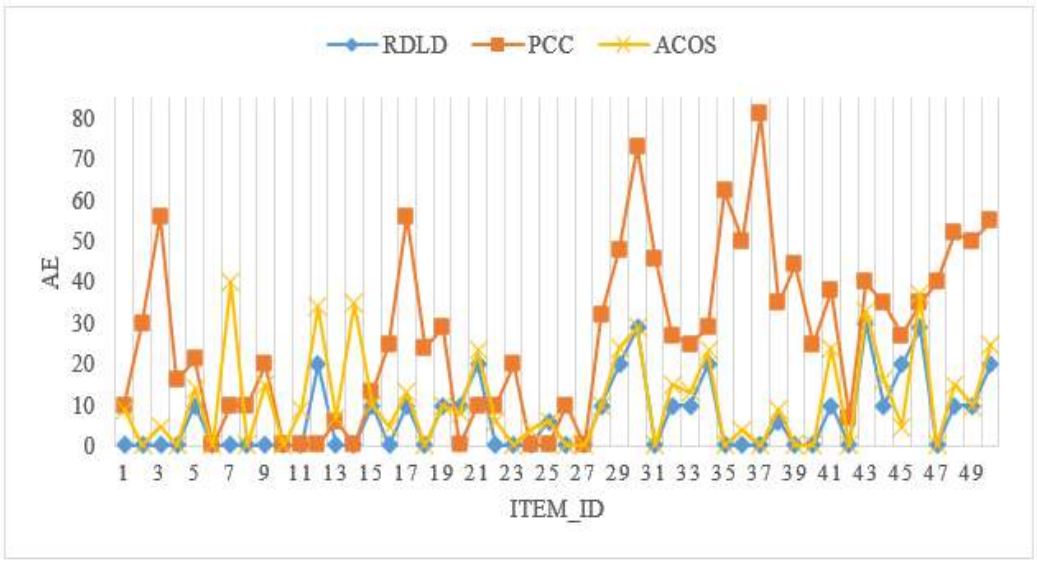

Figure 4.6. *AE for a user on Yahoo! music dataset*

From Figure 4.6, we can understand that the RDLD model maintained an advanced level in the AE value, as it scored a low value compared to the rest of the models. It ranged from 0 to 33/100 in the RDLD model, while the COS model came in the second stage with an AE value ranging from 0 to 40/100. Accordingly, the PCC model was ranked the third stage, as the AE values reached 81/100.

Thus, as average, the MAE is 0.537 in the RDLD model. It is 0.69, 0.76 in COS, and PCC, respectively (Figure 4.7)). Hence, the RMSE value of the RDLD model is 0.726. It is the best among the tested models, as illustrated in Figure 4.8.

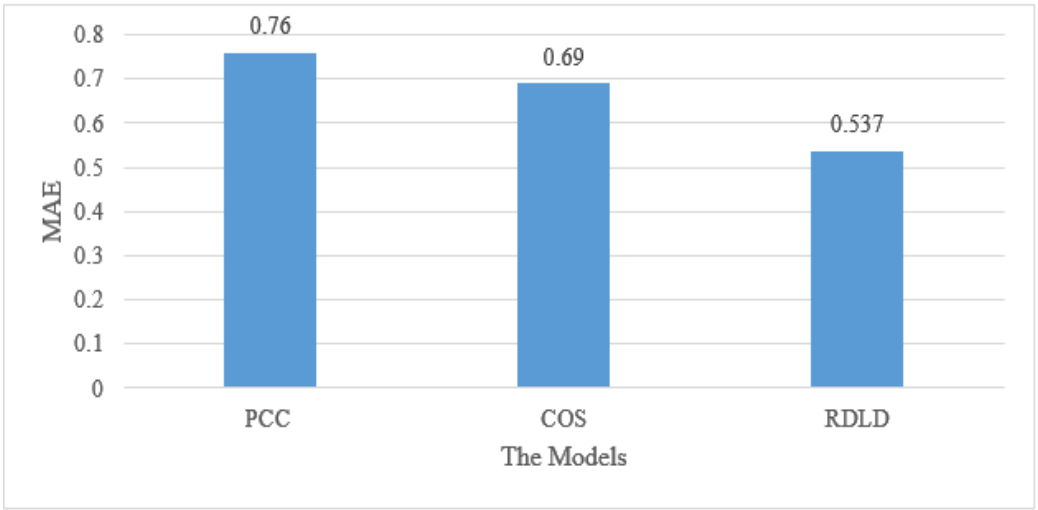

Figure 4.7. *MAE of models on Yahoo! music dataset*

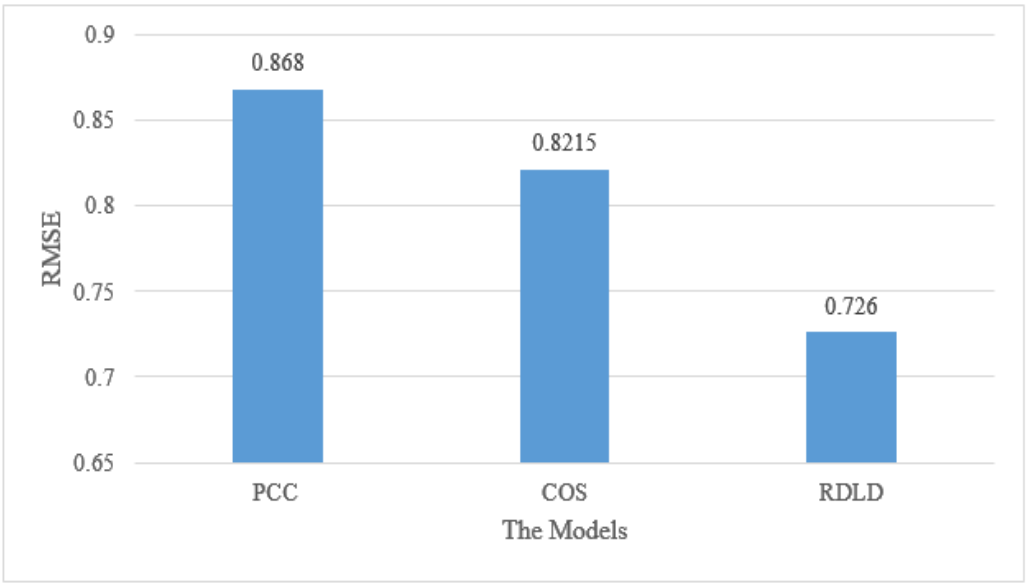

Figure 4.8. *RMSE of models on Yahoo! music dataset.*

In this part, we compared our model with some memory-based and model-based algorithms listed in ('Surprise Prediction_algorithms Package' 2020), including singular value decomposition (SVD), the CF, and other algorithms. These models contain different similarity measures and multiple prediction algorithms. In these algorithms, a fixed neighborhood size has been established that contains 50 neighbors.

Firstly, we applied these algorithms on the ML-20m dataset to find similarities between users in the experiments. Secondly, we applied different ratings prediction models on these similarities. Finally, we measured MAE and RMSE for tested users. Except for the RDLD model, which is based on dynamic k value, all models are based on K=50. The outcomes are represented in Figures 4.9 and 4.10.

We note that the RDLD gains better MAE than all other similarity measures. The second-lowest MAE value records by the K-NN Baseline algorithm is 0.644. At the same time, the Normal Predictor algorithm recorded the highest MAE value, MAE=1.15, followed by 0.701 and 0.685 by the Co-clustering and K-NN Basic algorithm (the traditional prediction algorithm in the CF algorithm), respectively. Similarly, with the RMSE evaluation metric, the RDLD surpasses all other similarity measures, followed by the SVD algorithm with RMSE=0.833.

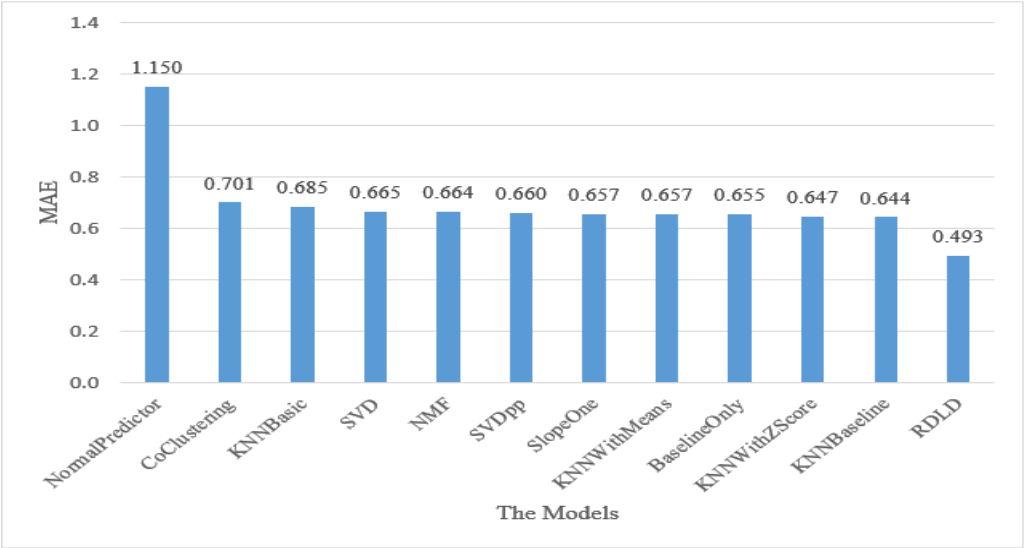

Figure 4.9. *MAE of other models on ML-20M dataset*

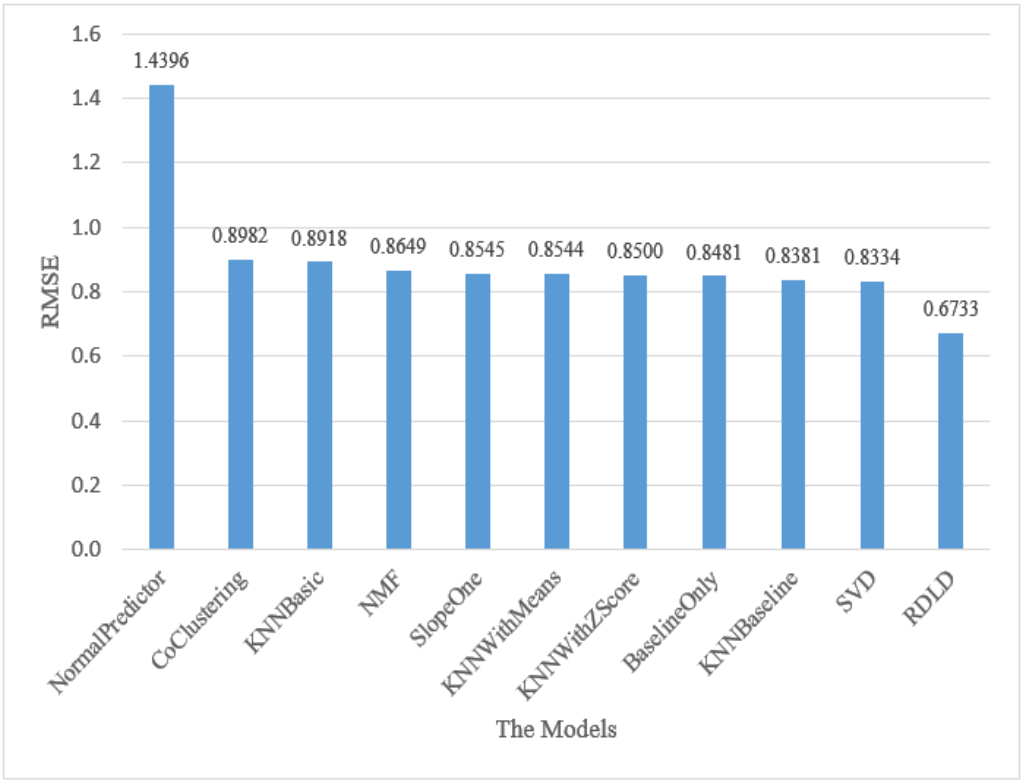

Figure 4.10. *RMSE of other models on ML-20M dataset*

Also, on the Yahoo! music dataset, the RDLD model scored first, with the lowest MAE and RMSE. Generally, the K-NN Basic algorithm recorded high MAE and RMSE, i.e., 0.895 and 1.251, respectively, as shown in the two figures (4.11 and 4.12).

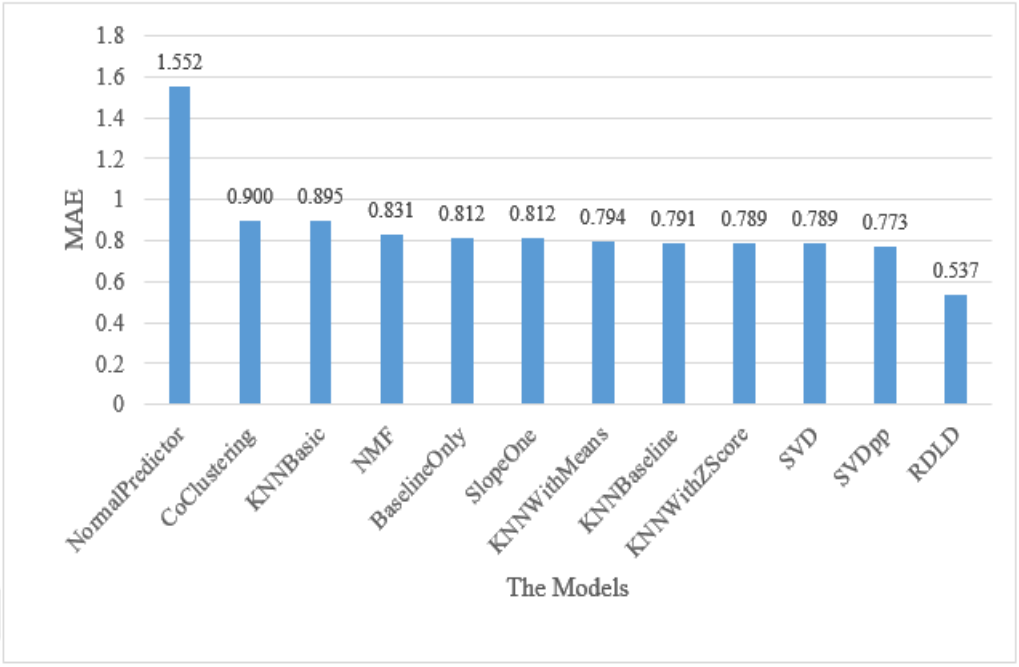

Figure 4.11. *MAE of other models on Yahoo! music dataset*

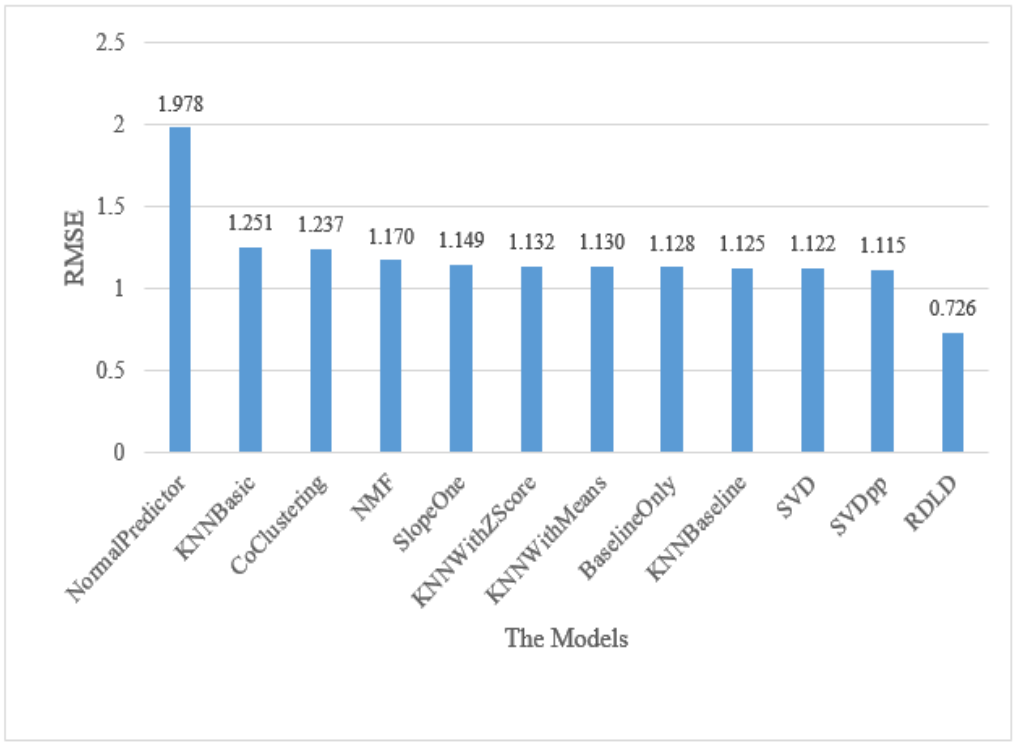

Figure 4.12. *RMSE of other models on Yahoo! music dataset*

Therefore, according to the MAE and RMSE evaluation criteria values, the RDLD model defeats some model-based algorithms such as SVD and NMF, and the memory-based algorithms such as K-NN Basic and K-NN baseline.

2- Experiments over cold-start user

Because the RDLD model was developed to improve recommendation quality in cold-start scenarios, we evaluated simulated new user cold-start conditions by allowing the similarity measure model to exercise a limited number of ratings for each user. The number of ratings included in the similarity calculation was increased from two to twenty. The prediction and MAE calculation were performed by RDLD, PCC, COS similarity measures, and the remainder algorithms on the selected datasets for each number of ratings. The RDLD, PCC, and COS model outcomes are decisive for the RDLD measure (Figures (4.13 and 4.14)).

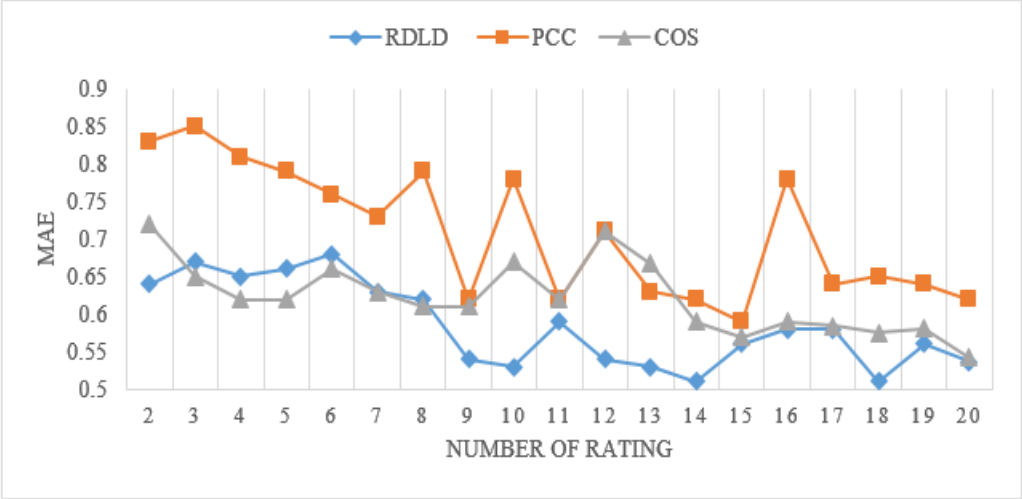

Figure 4.13. *MAE of new user cold-start experiments on ML-20M dataset*

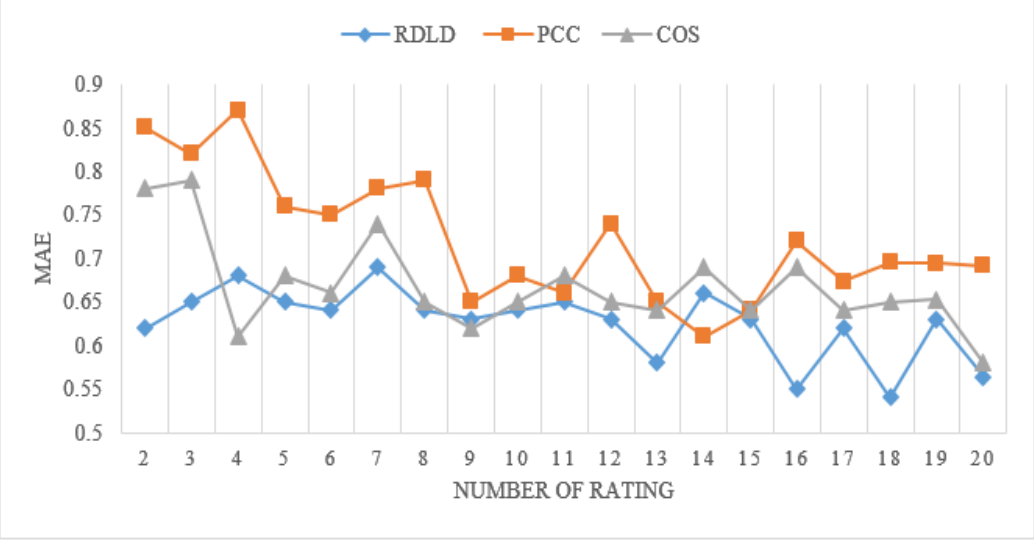

Figure 4.14. *MAE of new user cold-start experiments on Yahoo! Music dataset*

First, for both datasets, RDLD presents the best performance with the lowest MAE over several ratings trained.

Second, it is noticed that COS's performance regularly improves and approximates RDLD at some rating level; this is compatible with the result produced from the full rating analysis. In all steps, the RDLD model is followed by the COS for the best performance.

As a consequence of these findings, it can be concluded that the RDLD measure outperforms the other two similarity measures in cold-start scenarios. The RDLD model's dominance is proceeding, especially when the ratings are from 10 to 20 in both datasets.

For the remainder algorithms, the results are presented in Figures (4.15) and (4.16).

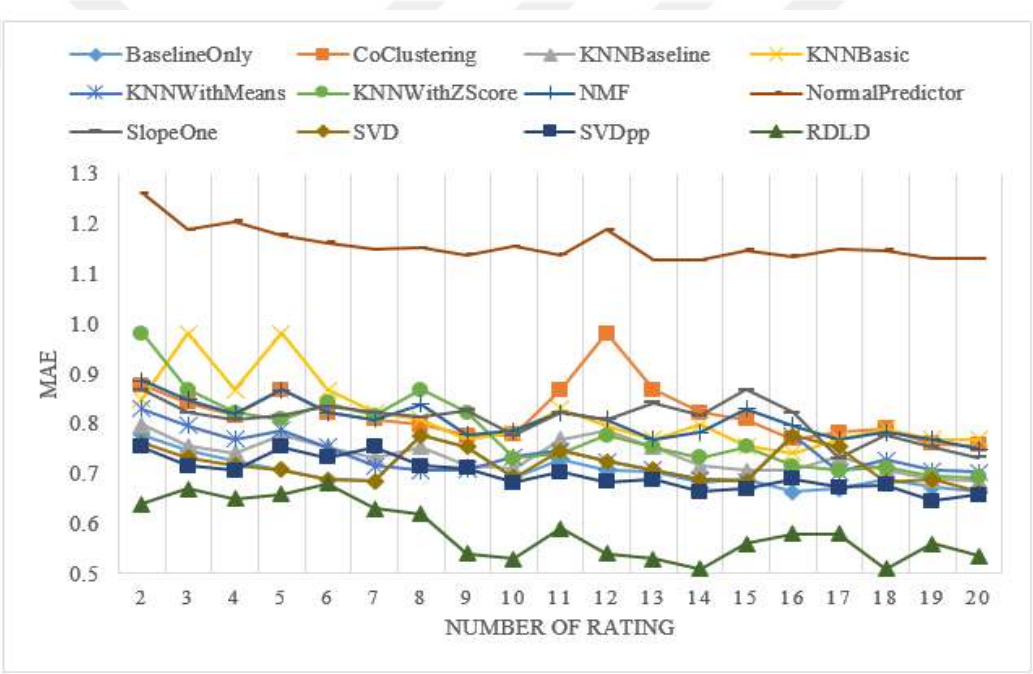

Figure 4.15. *MAE of new user cold-start experiments on Movie Lens dataset by a group of algorithms*

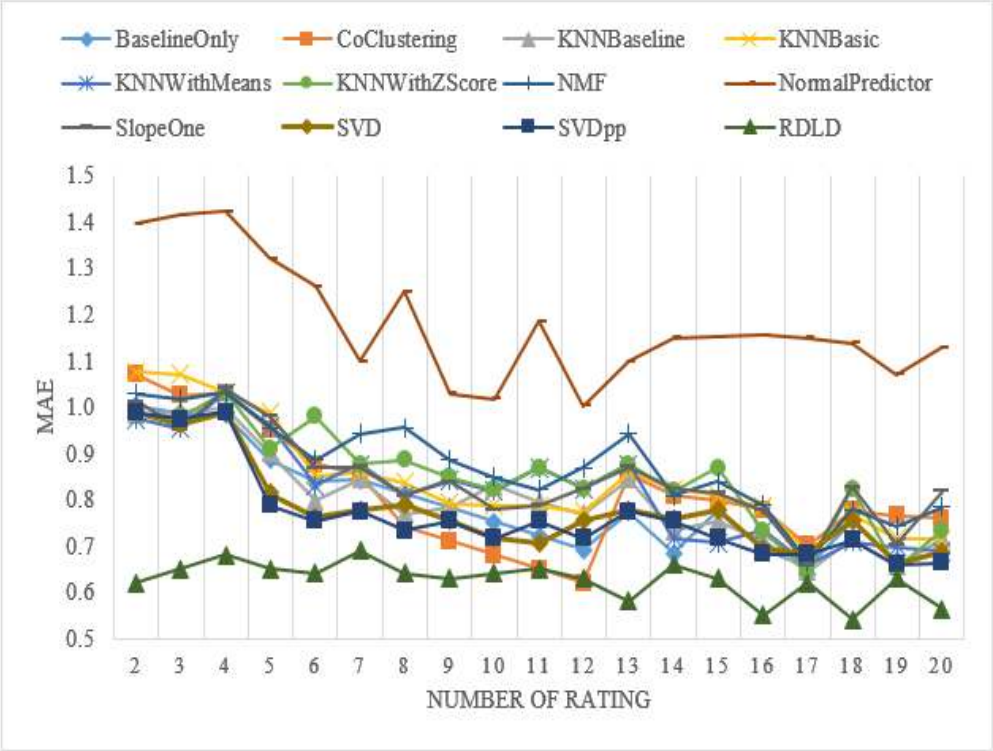

Figure 4.16. *MAE of new user cold-start experiments on Yahoo! Music dataset by a group of algorithms*

The RDLD model achieves the best performance. It outperforms other measures at all levels of multiple ratings by new users, implying that the RDLD model can be effective in most practical situations, even with a small number of ratings from new users.

3- Comparison to modern CF algorithms

To clarify the effect of the proposed similarity measure model in this research, this model is compared with the algorithms in the literature of (X. He and Jin 2019), (S. Li and Li 2020), and (Ayub et al. 2020). We compared all of the research using the same data set (ML-100k) and the same performance metric; the MAE. Therefore, we calculated the MAE value for the same dataset by our proposed model and compared it with the previous research; all details and results are listed in Table 4.5.

Table 4.5 shows how many K neighbors were tested in each research, the number of neighbors at which the best error value was recorded, and the indicated model's best MAE value.

Table 4.5. *Comparison of literature's details*

Literatures	K	Min MAE at	Min MAE
(Ayub et al. 2020) R_J Th<0.3	5 to 50	k=10	0.9
(Ayub et al. 2020) R_J_RPB	5 to 50	k=10	0.88
(S. Li and Li 2020)	10 to 50	k=10	0.78
(X. He and Jin 2019)	20 to 100	k=100	0.75
RDLD	Dynamic 1 to 50	0<k<8	0.538

From Table 4.5, the results illustrate the precedence of RDLD, not only in the MAE value but also in the number of neighbors it takes to predict the similar to correct rating; this indicates that the RDLD model has chosen the right neighbors.

4.4. Summary

Nowadays, most people own smartphones, tablets, and computers to access social network sites, shopping, and e-commerce sites. However, the proliferation of information and goods leaves people perplexed. Consumers must spend considerable time exploring for something they want. Even so, users may not be able to obtain the satisfactory results of their aspirations. Hence, it is helpful to have users' behavior tracking services available on social networks and online shopping sites. In this regard, a system has been used that provides customized services to users by analyzing their behavior, called a recommendation system (RS). It tries to identify a user's current favorites or anticipate a certain item's rating (Jain et al. 2015; Asanov 2011).

To summarize, experimental data indicate that the RDLD model outperforms all other models evaluated in this chapter. These experiments prove the RDLD effectiveness in identifies the similarities between users based on only common rating and small neighborhood size. Therefore, the RDLD and the applied rating prediction model succeed in selecting suitable neighbors and obtaining accurate rating predictions.

5. A MEASURE OF USER SIMILARITY BASED ON ITEM GENRE

Collaborative filtering is a method used by recommendation systems that predicts and suggests things of interest to the user. Because users' emotions vary, relying on user ratings for items may result in incorrect suggestions. As a result, it is critical to introduce a new similarity metric that improves suggestion accuracy even when consumers leave little reviews. Thus, this chapter offers a user similarity measure method that uses item genre as additional information to provide more accurate suggestions. This algorithm establishes a link between users based on item genre information, identifies the current user's most related neighbors in each genre, and generates a list of the active user's final nearest neighbors. They share his taste across all genres. Finally, it uses a defined prediction technique to forecast the active-user rating of goods.

Additionally, to offer a solution to the user cold-start problem by utilizing the "Ask-to-rate" technique/adaptive approach as a fundamental notion, the new user's chosen item genre is elicited. Then the top-k users who like the same genre are suggested to be the new user's nearest neighbors. To quantify accuracy, we propose two additional assessment criteria: rating level and user reliability at that rating level. We validate the suggested technique using real-world data sets. The empirical findings demonstrate that the proposed method outperforms several current collaborative filtering techniques in terms of projected rating accuracy, rating level, and inter-user reliability.

5.1. Introduction

Despite numerous studies and analyses on the user-user and item-item CF similarity measures, these measures did not infer sufficient similarity in some cases. Standard CF algorithms offer recommendations employing user' ratings for items regardless of the effect of many features on user similarity (Koren, Bell, and Volinsky 2009) and item similarity (B. Sarwar et al. 2001). It is essential to find a similarity measure based on the actual preferences of users rather than the users' rating value. Therefore, researchers are considering the effect of item characteristics on the recommendation accuracy of RSs (Choi, Ko, and Han 2012; Santos, Goularte, and Manzato 2014; Soares and Viana 2015). A model that mixes the item-based CF information content is introduced by (Q. Li and Kim 2003). They created a clustering method based on this mixed information. The K-mean clustering model and weighted deviations were used to determine the closest

neighbors, then calculated the rating prediction for unrated items. In (K. R. Kim and Moon 2012), the authors formed a special RS by analyzing the item genre relationship and user-preferred genres. The Pearson correlation coefficient (PCC) and clustering approaches are used to calculate genre similarity. Then, the model can suggest a recommended genre to the active user.

However, the CF algorithms suffer from several obstacles, and the popular ones are data sparsity and user cold-start. A new user with no rating is called "Pure new user". An increased number of new users and multiple less-active users joining up for e-commerce sites in all applications cause real difficulty for the existing recommendation algorithms (Al-Shamri and Bharadwaj 2008). At the same time, the problem with the cold item occurs by the lack of ratings for the new item (Ahn 2006; Middleton, Alani, and Roure 2002). Depending on CF, the user cannot receive any recommendations from the system (Rashid et al. 2002; Ren et al. 2008). Acceptable recommendation quality cannot be obtained when little or no information is available (Ahn. 2008).

Recent literature indicates that many RS' problems have been approached through various hybrid ways, including side information to determine users' interests and purchasing habits (C. Li et al. 2015). The study (H. N. Kim et al. 2010) presented a method based on exploiting collaborative tagging as supporting information to extract the users' tastes for the items. This study overcame the user cold-start and sparsity problems.

Regardless of many solutions proposed to solve the user cold-start and sparsity problems, such as (Subramaniyaswamy and Logesh 2017), (C. Y. Liu et al. 2018), and (Zare et al. 2019), most of these solutions rely only on user ratings rather than the actual user's preferences for items. Therefore, there is indeed a deficiency of features upon which users' similarity is determined. Similarity metrics should not be limited to users' ratings of specific things or matching their primary demographic data. There is a pressing need to examine additional and diverse features that adequately describe individuals across several domains. In this chapter, the fundamental objective is to create a new similarity measure by measuring the similarity between users based on preferred item genres. Furthermore, enhancing the prediction accuracy of RSs, and solving user cold-start problems.

5.1.1. Problem definition

According to (C. Y. Liu et al. 2018) research, users' preferences are not fixed and indicate fluctuations in their preferences for items. Hence it is not easy to find a similarity method that matches all user behaviors. Item genre-based for user similarity is not widely explored in the recommendation systems. Furthermore, the traditional similarity measures do not predict accurate recommendations for new users or items, and they are less effective for sparse datasets (Subramaniyaswamy and Logesh 2017; Zare et al. 2019). Initially, a precise similarity measure can be configured between users depending on the user's preferences for the item's genres. Our research experiments based on a movie lens (Harper and Konstan 2015) and yahoo music data sets looked at user preference for movie/ music genres. Some movies have multiple genres; as in Figure 5.1, two users, u1, and u2 are highly equivalent in their choices and ratings of action, fantasy, adventure, and horror. At the same time, they are different in their choices in the comedy and drama movie genre. Consequently, this is related to their personalities.

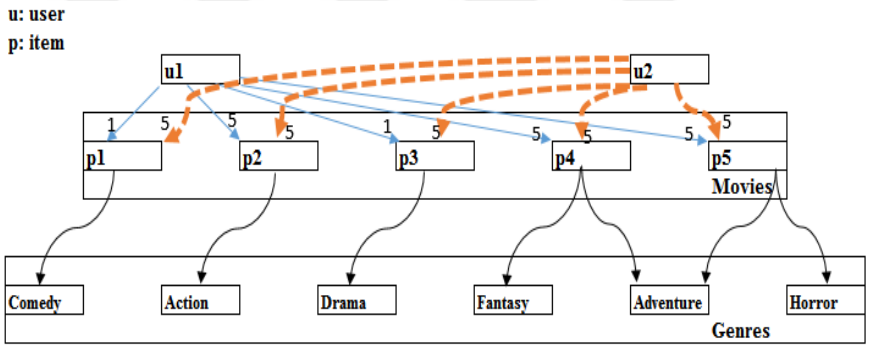

Figure 5.1. *Example of user preferences for movie genres*

Traditional similarity algorithms hold all u1 and u2 ratings collectively. Calculating their similarity shows that their relationship is medium because of their remarkable similarity in action, fantasy, adventure, horror movie genre, and weak similarity in comedy and drama. Hence, the results fail to show that their preferences on many movie genres are notably significant and reasonably weak on the other genres. Thus, the traditional similarity algorithms are incapable of representing the different choices between the multiple interest users. A similar rating might not mean similarity in actual preferences. Therefore, relying solely on user ratings may produce incorrect similarity rates between users or items. Watchers can frequently imagine the movie's story and spirit through its genre(s); hence, the movie's genre guides viewers' attention and helps them

decide whether or not to attend it (Choi, Ko, and Han 2012). Accordingly, extracting critical choices and contextual information from the users' profiles is essential to establishing vital similarity metrics among users. Therefore, we planned to solve this problem.

5.1.2. Novelty

Since most of the dataset is sparse (Qian et al. 2014; Y. Zhang and Song 2009), efficient solutions should avoid the sparsity problem and not rely entirely on user ratings. Therefore, to increase recommendation accuracy, dominate the user cold-start and data sparsity problems, a user-based similarity model according to the user's preference for the item genre has been suggested. The novelty of the proposed similarity measure is represented by measuring the user's average preference for a specific genre. Firstly, we calculated the average of the active user' ratings for items in his preferred genres; secondly, computing users' similarity by comparing the average user preferences across genres. Then, by specifying the target user's initial neighbors in each genre based on a certain similarity threshold with other users, we can create the target user's initial neighbors list. Then, extracting users who share the target user's tastes across all of his preferred genres, created the target user's final neighbors' list.

Besides, we follow the "Ask-to-rate" technique/ adaptive method (Nadimi-Shahraki and Bahadorpour 2014) for proposing the user cold-start solution. The novelty in this solution is that the system will ask the new user to answer the "What is your preferred genre" question during the registration process instead of forcing him to rate pre-defined items. According to the user answer, the system will select the highest average' users in that genre to be the nearest neighbors for this new user. The system thus begins creating a profile for this new user.

5.1.3. Research contribution

Unlike algorithms that solely rely on user ratings for items, we seek to develop a similarity measure that incorporates user interest and identifies user-suitable neighbors. Since specialists in the relevant item field handle the item genre information, the genre is regarded as reliable information used to identify user similarities and resolve user cold-start issues. The proposed technique computes users' average preference for each item genre and identifies nearest neighbors in each genre based on a user similarity threshold value. By intersecting user preferences per genre, it determines neighbors who share the

same taste across genres and considers them the ultimate neighbors for the current user. Additionally, we present a system to the new user by focusing on the "Ask-to-rate" technique/adaptive method (Nadimi-Shahraki and Bahadorpour 2014), distinct from previous approaches. The system inquiries about the new user's favorite genre instead of rating pre-defined items. We applied our method to real data sets and performed different experiments to verify the suggested method's accuracy and efficacy in picking neighbors and resolving the user cold-start problem. When compared to past baseline procedures, our novel approach consistently generates superior results.

This research aims to develop an effective user similarity metric depending on the user request for item genre and a solution to the user's cold-start issue. The essential contributions of this article are:

- Suggesting a model for measuring users' similarity according to their general preference for item genres;
- Introducing an "Ask-to-rate" methodology for resolving the new user problem;
- Proposing new evaluation criteria: level of predicted ratings and users' reliability;
- We prepared a detailed analysis to verify our presumption that the suggested model exceeds the current CF algorithms.

5.2. Proposed Model

A user-based relationship analysis model has been offered based on the item genres. The current CF-based approaches cannot suggest items for the users with acceptable accuracy since a data set sparsity is high. Therefore, they offer undesirable recommendations, in particular, for the user's cold-start situation. To rectify this difficulty, we aimed to define the relationship between users by exploiting the item genre information, which appears confident in robust relationship measurement even between cold-start users with reasonable error value in the sparse data set.

As explained before, examining only users’ ratings to find similarities between users does not consistently deliver accurate recommendations. Two users might rate the same movie with the same rating for multiple reasons, which differ from one user to another, which may be due to his desire for the movie genre or because he prefers the

movie's actor and other reasons. Therefore, the similarity method must expose the actual user preferences.

5.2.1. User-based similarity measure using item genres information

This section includes a user-based similarity measure model, defined from users' preferences for item genres. In this model, the classifications of users have been prepared according to their preferences for item genres. Then, the similarity between users is determined according to the average user rating for the item genres, so it is called the user-genre similarity model (UGSM), as illustrated in Figure 5.2.

As a result of classifying users according to their favorite item genres, we construct Table 5.1, representing a sample of five users and their preferred genres. Assume that U= {u1, u2... un} represents the set of users, and G= {g1, g2... gs} represents the collection of genres, and p= {p1, p2… pm} is a set of common items. The item genres in Table 5.1 refer to the movie genres identified in the movie lens dataset (Harper and Konstan 2015), and the values represent the average users rating to their preferred genres.

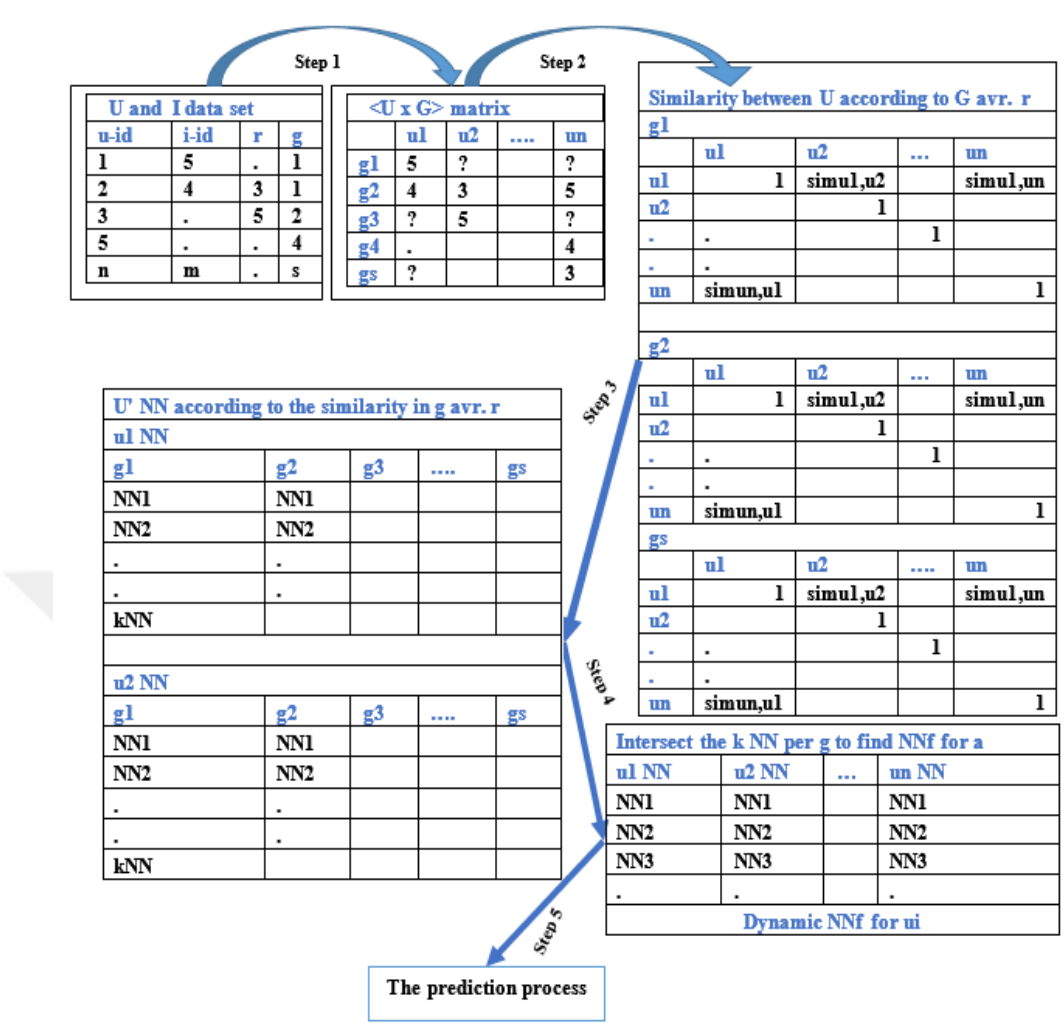

Figure 5.2. *Proposed model flowchart for regular users*

Table 5.1. *Users (u) and their mean rating on preferred Item genres (g)*

g name	u1	u2	u3	u4	u5
g1: Action|Comedy|Sci-Fi|Thriller	4	5	3	5	3
g2: Action|Thriller	3	5	0	3	3
g3: Adventure|Animation|Children	0	4	0	4	3
g4: Comedy	3.5	4	5	3	5
g5: Adventure	5	0	3.5	4	5

From Table 5.1, it is clear that an item may be related to multiple genres, and the user preferences differ from one genre to another. Selecting an active user "a" to a particular genre means the user prefers that genre. Therefore, there are remarkable opportunities for two users to be highly similar when they like the same genres. The essence of the memory-based CF algorithms is calculating the actual similarity between

entities for selecting proper neighbors. Thus, in addition to the rating prediction phase, the proposed similarity metric involves four significant steps:

Step 1: Building a genre rating <u×g> matrix from the master data set, including (u-id, i-id, r, i-g) by calculating the user preference average per item genre (g).

Step 2: Finding the similarity between the users' preferences per genre using Eq. (5.1). A slight difference indicates a significant similarity.

$$diff(a, u_i) = | \, avr\, r_{a,gj} - \, avr\, r_{ui,gj} \, | \;\Leftrightarrow\; a \text{ , and } ui \; preferred \;\; g_j \quad (5.1)$$

Where:

$i = 1 \ldots, n$, and $j = 1 \ldots, s$.

$diff(a, u_i)$: represents the degree of difference between a and u preference for the same genre g. A smaller value means more significant similarity between users.

$avr\, r_{a,gj}$: represents the average ratings of a to g_j.

$avr\, r_{ui,gj}$: represents the average ratings of ui to g_j.

Users similarity is calculated as the absolute difference between users rating' average for the same genre.

Step 3: Finding the nearest neighbors (K-NN) for each user per genre by arranging the difference values in the preferences rate in ascending order.

Step 4: Find the final nearest neighbors (NNf) for the active user by intersecting K-NN groups in the previous step to get the neighbors who share all their preferences with the active user, according to Eq. (5.2).

$$a_{NNf} = \; a_{kNN,g1} \cap a_{kNN,gj} \quad (5.2)$$

Where:

a_{NNf}: It is the final $a's$ NN.

$a_{kNN,g1}$: It is the K-NN for a in the first common genre; $g1$.

$a_{kNN,gj}$: It is the K-NN for a in the j[th] common genre.

In fact, as display in Table 5.1, if two users are similar to each other in their preference for a genre, they may differ for different genres. Therefore, the similarity model must carefully choose a similar neighbor to a in his preferences for all genres. Accordingly, the UGSM includes an intersection process among the first K-NN in all genres to find a's closest neighbors similar to a in all preferences.

Each a will have several neighbors, which differs from another a. The reason for that is, the intersection process will keep users who are similar with a in all genres. Thus, the results of the intersection process among users vary from one a to another. For example, the first a might get 40 NNf, while the second a might get only 35. If the similarity value among a and many neighbors is equal, to form the a_{NNf}, these neighbors are arranged in descending order according to the number of common genres shared with a. To maintain the most significant number of neighbors sharing one or more genres with the active user, the proposed model includes the following process:

If $a_{kNN,g1} \cap a_{kNN,gj} = \{\}$, then, $a_{NNf} = a_{kNN,g1} \cap a_{kNN,gj-1}$ else

$a_{NNf} = a_{kNN,g1}$.

Step 5: The active user rating prediction process.

To recommend the correct items to the active user, the final step in the proposed model is the active user rating prediction. The proposed model relied on the prediction model proposed in (J. K. S. Al-safi and Kaleli 2021); as illustrated in section 3.3.2. His final neighbors' average aggregate rating differences are ordered in descending order to establish the proper neighborhood's ranking.

5.2.2. User cold-start proposed solution

The second part of the proposed model includes a solution to the user's cold-start problem. Followıng the "Ask-to-rate" technique/ adaptive method (Nadimi-Shahraki and Bahadorpour 2014). We presume that when a new user registers with the system, the system asks them to choose their favorite genre. The system then recommends the top-k individuals who rated the same genre as this user's nearest neighbors.

Assume that v_{gj} represent column vectors for each gj. Each vector contains average user ratings who preferred gj i.e. $avr\ r_{u,gj}$ sorted in descending order:

$$v_{gj} = [\, avr\, r_{u,gj}]^t$$

Then, assume that $Index\ v_{gj}$ represents column vectors for each v_{gj}. $Index\ v_{gj}$ Contain the index order of users who preferred gj:

$$Index\ v_{gj} = [index1\ u\ , index2\ \ u, index3\ u, \dots]^t$$

The K-NN for the new user is the first K-indexed user from $Index\ v_{gj}$; who preferred the same genre. The items preferred by these neighbors will be suggested to the new user.

5.2.3. An illustrative example of the proposed algorithm

For elucidation, the following example explains the UGSM similarity measure model: Assume that we have five users (u1, u2, u3, u4, and u5) and five genres (g1, g2, g3, g4, and g5). Based on Table 5.1 and Eq. (5.2). The differences in values between the five users are shown in Table 5.2. The symbol "-" denotes the value of the difference between two users on the uncommon genre.

Table 5.2. *Differences between 5 users according to 5 preferred genres*

g1		**u1**	**u2**	**u3**	**u4**	**u5**
	u1	0	1	2	1	2
	u2	1	0	2	0	2
	u3	2	2	0	2	0
	u4	1	0	2	0	2
	u5	2	2	0	2	0

g2		**u1**	**u2**	**u3**	**u4**	**u5**
	u1	0	2	-	0	0
	u2	2	0	-	2	2
	u3	-	-	0	-	-
	u4	0	2	-	0	0
	u5	0	2	-	0	0

g3		**u1**	**u2**	**u3**	**u4**	**u5**
	u1	0	-	-	-	-
	u2	-	0	-	0	1
	u3	-	-	0	-	-
	u4	-	0	-	0	1
	u5	-	1	-	1	0

g4		**u1**	**u2**	**u3**	**u4**	**u5**
	u1	0	0.5	1.5	0.5	1.5
	u2	0.5	0	1	2	1
	u3	1.5	1	0	2	0
	u4	0.5	2	2	0	2
	u5	1.5	1	0	2	0

g5		**u1**	**u2**	**u3**	**u4**	**u5**
	u1	0	-	1.5	1	0
	u2	-		-	-	-
	u3	1.5	-	0	0.5	1.5
	u4	1	-	0.5	0	1
	u5	0	-	1.5	1	0

Then, we find the first 3-NN for *u1*, like in Table 5.3.

Table 5.3. *First 3-NN for u1*

g1	**g2**	**g4**	**g5**
u2	u4	u2	u5
u4	u5	u4	u4
u5	u2	u5	u3

Accordingly, from Table 5.3, for u1 first 3-NN in g1 are u2, u4, and u5. In g2 are u4, u5, and u2, and so on. Therefore, based on Eq. (5.3), the final list for u1 NN (u1 NNf) contains only u4 and u5. In the prediction process, the predicted rating will be based on u4 firstly because the average difference between u1 and u4 is 0.625, with u5 is 0.87. However, if there is an unrated item in the u4 profile, the prediction operation will be based on u5 ratings.

The following example explains our proposed solution for the user cold-start problem: the system asks the user to specify his preferred genre; assume that the new user chooses the "Drama" genre;

$$v_{Drama} = \left[\, avr\ r_{u,Drama}\right]^t ;\ v_{Drama} = [5,5,4,4,4,3,3,3]^t$$

And

$$Index\ v_{Darma} = [u10\ , u8\ \ , u25, u22, u100, u52, u84, u2]^t$$

Which mean $avr\ r_{u10,Drama} = 5,\ avg\ r_{u8,Drama} = 5,\ \ avr\ r_{u25,Drama} = 4$; and so on. Consequently, $new\ user_{kNN}$=$[u10\ , u8\ \ , u25, u22, u100, u52, u84, u2]^t$. Then, the system suggests the elements that these neighbors preferred to the new user.

5.2.4. Proposed evaluation metrics

- Predicted rating level (P. L.)

Since we endeavor to improve the accuracy of RSs, we suggested a new evaluation measure named predicted rating level determination, which divides the ratings into:

- The positive level (*POS*) is represented when the rating *r > (max/2)*; where "max" means the maximum score rating in the data set.
- The average level (*AVG*) when *r= (max/2)*; and
- The negative level (*NEG*) when *r< (max/2)*.

We suppose that if the predicted and original ratings are both at the same level, the proposed model's outputs are accurate. The model can select an appropriate neighborhood.

- Communities reliability level (Rel. L.)

We strive to quantify community reliability. As a result, we presented a new measure of reliability based on the ratio of common ratings with a similar rating level between the active user and his neighbors to the total number of common ratings. A high value indicates better reliability. The reliability can be obtained from Eq. (5.3):

$$Reliability\ (a, u_i)\ = \frac{\#\ I(a,u_i)_{rLa=rLui}}{\#\ I(a,u_i)} \qquad (5.3)$$

Where:

$\#\ I(a,u_i)_{rLa=rLui}$: The number of common item *I* between a and his neighbors u_i, which are at the same rating level rL; *i = 1..., K-NN.*

$rLa, rLui$: The rating level of a and ui.

5.3. Experiments

The experiments were performed on real data sets to examine the accuracy of the outcomes when employing the suggested UGSM. In this research, we relied on the prediction method presented in (J. K. S. Al-safi and Kaleli 2021) for rating prediction. The experiments included evaluating the accuracy, the level of the predicted rating, and reliability between users, in addition to the comparison of UGSM to current CF algorithms.

5.3.1. Data sets

We relied on an offline analysis approach based on previously collected data from websites (Beel et al. 2013). We used two offline data sets: Movie lens (ML-20M) (https://grouplens.org/datasets/movielens/20m (2 February 2021)) (Harper and Konstan 2015) and R2-yahoo! Music user ratings of songs with song attributes, version 1.0 data sets (https://webscope.sandbox.yahoo.com/catalog.php?datatype=r&did=2 (15 May 2021)). ML- 20M it included two separate files: the first file consisted of (User id, Movie id, Rating) and the second file consisted of (Movie id, Movie title, Movie genre). Movies are classified into 20 genres. The R2-yahoo! Music dataset represented music preference

data for songs from the Yahoo site. It included multiple files: the first file contained (User id, Song id, rating), the second file included (Song id, Music genre id), and the last file included (Genre id, Genre name). We used the data in the "ydata-ymusic-user-song-ratings-meta-v1_0/train-1.txt" file.

In both datasets, the files were merged to implement the proposed algorithm. We needed a user id, movie/song id, movie/music genre id, and user rating for the movie/song. Furthermore, we compared our findings to previous researchers' findings by utilizing additional datasets (ML1 and ML-1M). The sparsity level was obtained using Eq. (3.8).

In brief, our experiments were based on the first 1500 users who had the highest number of ratings (more than 100 ratings). The dataset details are shown in Table 5.4.

Table 5.4. *Data sets Detail*

Dataset	**Scale**	**U**	**I**	**R**	**No. g**	**Spar (%)**
ML1	0.5–5 St	671	9125	100,000	18	98.36
ML-1M	0.5–5 St	6040	3952	1,000,209	20	95.8
ML-20M	0.5–5 St	7120	27,278	1,048,575	20	99.0
Adapted ML-20M	0.5–5 St	1500	7500	676,312	19	93.96
R2-yahoo! Music	1–5 St	2829	127,217	1,048,572	58	99.7
Adapted R2-yahoo! Music	1–5 S	2663	17,327	147,223	57	99.5

Since the proposed algorithm depends on genre information, we describe the genre information in Tables 5.5 and 5.6 for the ML-20M dataset and Table 5.7 for R2-yahoo! Music dataset. Table 5.5 shows the genre names and the number of movies that belong to them. The drama genre was the most common, and there were 13,344 drama movies. The comedy genre came in second, and there are 8374 movies. Thus, this dataset contained 246 movies belonging to the "unknown" genre, which we excluded from the dataset in the experiments.

Table 5.5. *Genre name in ML-20 M dataset and number of movies per genre*

No.	g Name	# Movies	No.	g Name	# Movies
1	Drama	13,344	11	Mystery	1514
2	Comedy	8374	12	Fantasy	1412
3	Thriller	4178	13	War	1194
4	Romance	4127	14	Children	1139
5	Action	3520	15	Musical	1036
6	Crime	2939	16	Animation	1027
7	Horror	2611	17	Western	676
8	Documentary	2471	18	Film-Noir	330
9	Adventure	2329	19	Unknown	246
10	Sci-Fi	1743	20	IMAX	196

Table 5.6 shows the number of genres and the number of movies in that genre. The movie may belong to more than one genre. A total of 10,829 movies belonged to a single genre, while 8800 movies belonged to the complex genre (i.e., contained more than one genre). Some movies belonged to five genres.

Table 5.6. *Several g and the movies of that g in ML-20M dataset*

# g	# Movies	# g	#Movies
1	10,829	6	83
2	8809	7	20
3	5330	8	5
4	1724	9	-
5	477	10	1

Table 5.7 shows the genre name in the R2-yahoo music dataset and the number of songs that belong to it. Most songs have an unknown “un” genre, where there are 109,890 “un” genre songs. Since we did not use any item from the “un” genre in the experiments of the proposed algorithm, the rock genre was first, where there were 7013 songs. In this dataset, each song belonged to only one genre. Thus, this dataset contained 58 genre groups of a single size. Note that “S” represents a song.

Table 5.7. *Genre name in R2-Yahoo Music dataset and number of songs per genre*

No.	g Name	# S	No.	g Name	# S	No.	g Name	# S
1	Unknown	109,890	21	Adult Contemporary	75	41	Indie Rock	16
2	Rock	7013	22	Blues	73	42	Alt-Country	12
3	Pop	2776	23	Metal	59	43	Movie Soundtracks	12
4	R&B	2122	24	Pop Metal	58	44	R&B Gospel	10
5	Country	1118	25	Shows ; Movies	58	45	Hard Rock	8
6	Rap	920	26	Easy Listening	58	46	Techno	7
7	Classic Rock	443	27	Vocal Jazz	57	47	Holiday	6
8	Comedy	400	28	Industrial Rock	42	48	Gospel	6
9	Folk	228	29	Folk-Pop	37	49	Modern R&B	4
10	Jazz	225	30	Disco	32	50	Early Blues	3
11	Reggae	213	31	World	30	51	Traditional Folk	2
12	Latin	157	32	Ambient Tech	23	52	Minimal Techno	1
13	Mainstream Dance	145	33	Soft Pop	23	53	Mainstream Pop	1
14	Classic R&B	136	34	Death Metal	23	54	New Wave	1
15	Religious	124	35	Classical	23	55	Indie Pop	1
16	Speed Metal	122	36	New Age	21	56	Country Comedy	1
17	Modern Rock	87	37	Vocal Standards	21	57	Lounge	1
18	Christmas	84	38	Modern Blues	18	58	Orchestral	1
19	Electronic/Dance	79	39	Funk	17			
20	Adult Alternative	77	40	Punk	17			

5.3.2. Evaluation metrics

Evaluation metrics are an essential part of assessing a prediction in machine learning models. Since the introduced method can produce a forecast of ratings, MAE was used as a metric of algorithm performance. Additionally, we used the proposed evaluation metrics: level of predicted rating and the reliability between users that are described in section (5.2.4).

5.3.3. Experimental setup

We randomly divided the Adapted ML- 20M and the Adapted R2- Yahoo! Music dataset to 20% of users' rating as a test set to evaluate the suggested algorithm and 80% train set to train it. Then, we randomly select 100 ratings from each test user and replaced

their ratings with 0. In the rest of the paper, we call the adapted ML-20M and adjusted R2 - Yahoo! Music data sets ML-20M; R2 - Yahoo! Music, respectively.

5.3.4. Experimental procedures, evaluation, and discussion

We conducted three types of experiments with users by UGSM: users with a high number of ratings (i.e., regular users), users with 20 or fewer ratings, and users without ratings (i.e., pure cold-starting users). We described, evaluate, and discuss the results of each type separately.

- **Type 1: Experiments of regular users:**

Experiment 1: Determining the K-NN and minimum similarity

The proposed algorithm contains essential parameters: the active user's initial nearest neighbors in each genre (K-NN) and the final nearest neighbors for the active user in all genres (*NNf*). We tested choosing the optimal K-NN and minimum similarity (Avr. min. sim.), i.e., similarity threshold between the active user and his *NNf* list, to obtain optimal performance and examine changes to the accuracy of the proposed algorithm. Thus, different values of K-NN using UGSM, PCC, and COS, from 20 to 100 in increments of 5 per stage for both data sets. We measured the algorithms' accuracy according to MAE. Figure 5.3 shows the result of the ML-20M data set.

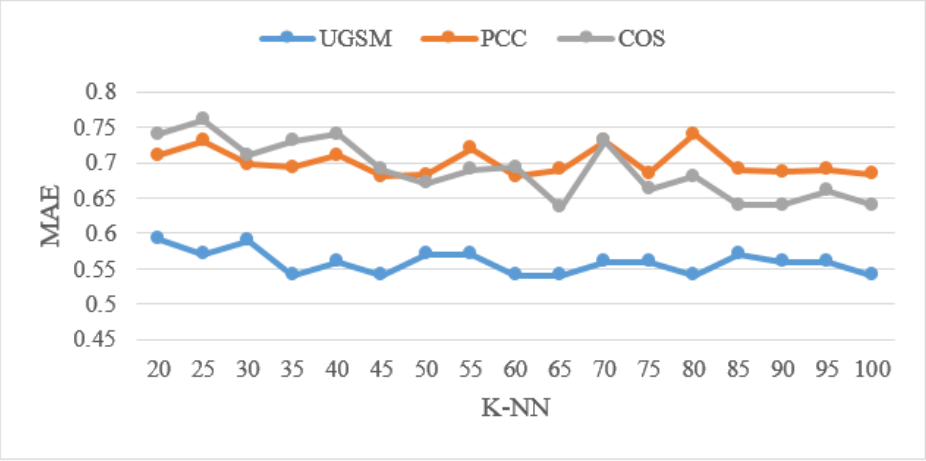

Figure 5.3. *MAE for different K-NN sizes on ML-20M by UGSM and other models. The x-axis is the K-NN size, and the Y-axis is the MAE value*

It is seen that the accuracy of UGSM prediction is better when K= 35, where MAE=0.54. Therefore, in the rest of the experiments, the value of K is fixed to 35. For comparison, PCC scored the lowest MAE=0.68 when K=60, and COS recorded

MAE=0.637 when K=65. In the UGSM, the "avr min sim" was derived using the following Eq. (5.4):

$$w\ avr.min.sim = 100 * \left(1 - \frac{\max\left(avr.diff(a_{gj}, u_{i,gj})\right)}{\max r}\right) \Leftrightarrow \ u_i \in NNf_a \text{ list} \quad (5.4)$$

Where:

NNf_a: represent the final NN for a;

$w\ avr.min.sim$: is the weight of average minimum similarity between a and NNf_a; i=1, 2,…, NNf_a, and j=1, 2…# of common g between a and NNf_a,

In this model, we assume that the weight of NNf_aon K-NN ($w.NNf_a = (\#NNf_a)/(\#\ K - NN)$) has a significant role in enhancing the prediction accuracy by finding the proper neighbors. Therefore, we chose the optimal value for similarity according to the lowest MAE value and the suitable size of NNf_a. The details are shown in Table 5.8.

Table 5.8. *MAE, NNf, and w min sim of adaptive ML-20M data set*

K-NN	W. NNf	W min sim	MAE	K-NN	W. NNf	W min sim	MAE
20	6/20=0.3	0.4	0.592	65	29/65=0.44	0.2	0.54
25	11/25=0.44	0.3	0.57	70	33/70=0.47	0.4	0.56
30	12/30=0.4	0.4	0.59	75	33/75=0.44	0.3	0.56
35	**22/35=0.62**	**0.5**	**0.54**	80	33/80=0.41	0.4	0.54
40	22/40=0.55	0.3	0.56	85	35/85=0.41	0.3	0.57
45	24/45=0.53	0.3	0.54	90	39/90=0.43	0.4	0.56
50	24/50=0.48	0.3	0.57	95	41/95=0.43	0.5	0.56
55	26/55=0.47	0.5	0.57	100	41/100=0.41	0.3	0.54
60	27/60=0.45	0.3	0.54				

From Table 5.8, it is notable that when *K*= 35, the W.NNf$_a$ is distinct and higher than the rest of the neighbors recorded when *K* was higher or less than 35. W.avr min sim was more significant than the remaining values, here w avr min sim=0.5, with it, the minimum MAE is recorded. PCC and COS have an avr min sim=0.2 and 0.23, respectively, with their ideal neighbors' size. In comparison, the UGSM model picks up suitable neighbors with less neighborhood size.

For the R2-yahoo music data set, Figure 5.4 declares the result of prediction accuracy of UGSM, PCC, and COS.

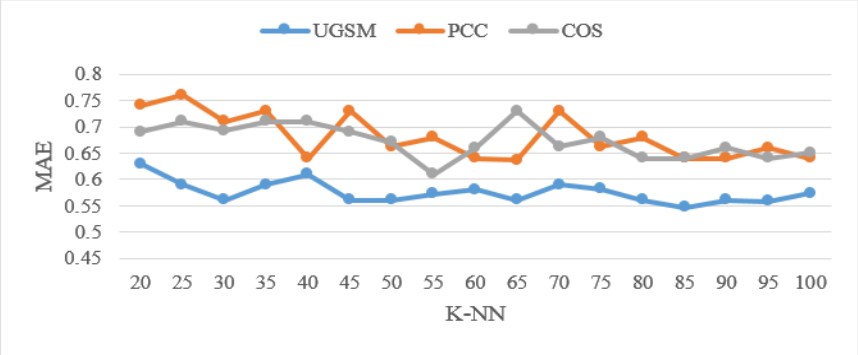

Figure 5.4. *MAE for different K-NN sizes on R2-yahoo music data set by UGSM and other models. The x-axis is the K-NN size, and the Y-axis is the MAE value.*

The accuracy of UGSM prediction is better when K = 45, where MAE=0.56. Therefore, in the rest of the experiments, the value of K is fixed to 45. While PCC has the lowest MAE=0.63 when *k*=65. However, COS has MAE=0.61 when *k*=55. Table 5.9 contains the values of avr min sim for the R2-yahoo music data set.

Table 5.9. *MAE, NNf, and w min sim of adaptive R2-yahoo music data set*

K-NN	W. NNf	W.min sim	MAE	K-NN	W. NNf	W.min sim	MAE
20	8/20=0.4	0.5	0.63	65	28/65=0.43	0.52	0.56
25	12/25=0.48	0.4	0.59	70	28/70=0.4	0.4	0.59
30	12/30=0.4	0.4	0.56	75	32/75=0.42	0.32	0.582
35	12/35=0.34	0.63	0.59	80	33/80=0.41	0.46	0.561
40	14/40=0.35	0.4	0.61	85	33/85=0.39	0.38	0.546
45	**23/45=0.51**	**0.54**	**0.56**	90	35/90=0.39	0.5	0.56
50	23/50=0.46	0.451	0.56	95	37/95=0.39	0.4	0.558
55	25/55=0.45	0.56	0.572	100	37/100=0.37	0.4	0.573
60	27/60=0.45	0.541	0.58				

From Table 5.9, it is distinguished that when K= 45, the W. NNf is high, and w avr min sim=0.54. While PCC and COS have an avr min sim=0.18, and 0.21, respectively, and greater MAE with their typical neighbors' size. By comparing UGSM to the PCC and COS similarity measures in both datasets, the results demonstrate that the UGSM captures better similarity, with less neighborhood size and small MAE. This result is a simple indication of UGSM's ability to choose the correct neighbors.

Experiment 2: Prediction accuracy

This experiment contains three parts:

The first part assessed the UGSM method's performance according to predicted ratings' accuracy by comparing it with PCC and COS. At the test stage, models produce a list of predicted ratings by utilizing the test datasets. We compared the predicted ratings to the original ratings in terms of MAE, predicted ratings' level, and reliability between users. All results are displayed in Figure 5.5.

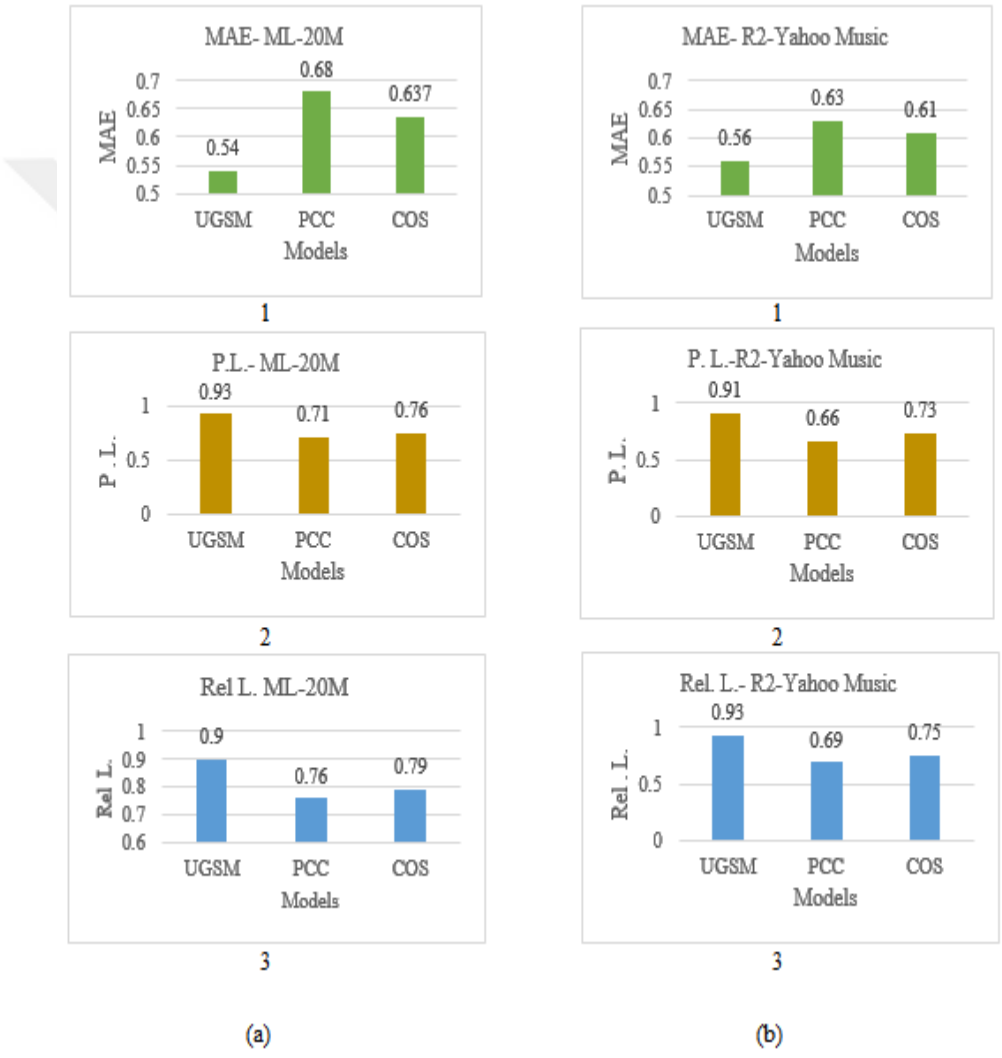

Figure 5.5. *Comparison of UGSM and traditional similarity measures according to (1) MAE; (2) P L; (3) Rel L; for (a) ML-20M data set, and (b) R2-Yahoo Music data set.*

From the results in Figures (5.5 a, and b), it is clear that the UGSM exceeds traditional similarity measures in both datasets since it produced an exceptional accuracy according to all tested evaluation metrics. When the K is at the optimal value in all models, UGSM's accuracy is significantly better than PCC and COS; firstly, in (Figure

5.5 a1) the result confirms that the UGSM decrease the MAE to 14.0 % against PCC, and 10.0% against COS for ML-20M data set. Secondly, (Figure 5.5 a2) regarding the P.L. of UGSM, it is 22.0% and 17.0% better than PCC and COS, respectively. Finally, (Figure 5.5 a3) by applying the Rel L metric, the result also continues confirming UGSM progress. It increases the Rel L between users by 14.0% and 11.0 % on PCC and COS. Anyway, the same progression is obtained for the R2-yahoo music data set.

In the second part, we compared UGSM with the modern CF algorithms in CF-SV by (Z. Su et al. 2020), User-MRDC by (Ai et al. 2019), and IOS by (X. S. He et al. 2015). These studies are used the ML1 data set and MAE as a performance metric. Accordingly, we measured the MAE value for the same data set by the UGSM. We compared our findings to the results obtained in the study (Z. Su et al. 2020), where 3 to 150 neighbors are used. Table 5.10 shows the K neighbors tested in each algorithm, the number of neighbors the best MAE recorded, and the best MAE obtained by the indicated algorithms. Figure 5.6 shows the results.

Table 5.10. *Comparison of UGSM' MAE to state-of-art algorithms.*

State- of-art algorithms	K	Min MAE at	Min MAE
CF-SV (Z. Su et al. 2020)	3 to 150	k= 20	0.685
IOS (X. S. He et al. 2015)	3 to 150	k= 50	0.73
User-MRDC (Ai et al. 2019)	3 to 150	k=30	0.698
UGSM	3 to150	0<k<=40	0.547

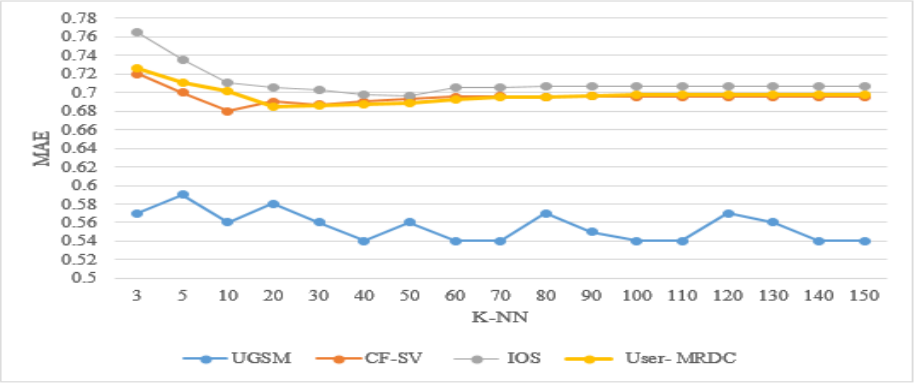

Figure 5.6. *Comparison of the UGSM accuracy to the state-of-the art algorithms for the ML1 data set. The X-axis is the k-NN, and the Y-axis is the MAE value.*

From Table 5.10, it is striking that UGSM gives the least MAE with a dynamic number of neighbors. It is clear that the UGSM model has better prediction accuracy than compared algorithms in all neighbors' sizes. Since the UGSM does not directly depend

on user ratings in selecting neighbors, its MAE is not affected much when changing its size.

In the third part, we compared the UGSM to the proposed algorithm in (L. Chen et al. 2021) "UW" algorithm, which is compared to a variety of other approaches, including MSD, COS, Jaccard, PIP, and CJMSD. They applied the experiment on the ML-1M data set. Thus, we used their data set methodology on the UGSM to compare with them. Figure 5.7 demonstrates the MAE values for the ML-1M data set for UGSM and several methods; K from 10 to 50, with increments of 5 per step.

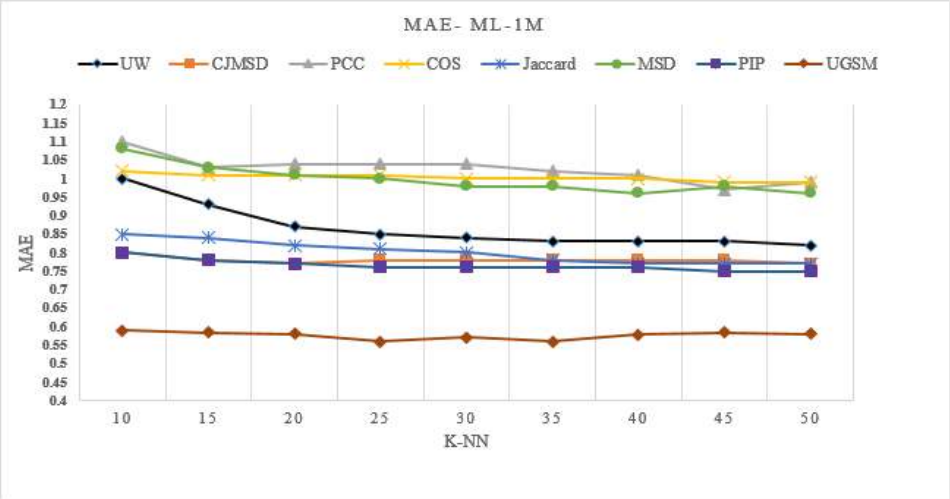

Figure 5.7. *MAE of the UGSM and the state-of-art studies for ML-1M data set. The X-axis is the k-NN, and the Y-axis is the MAE value.*

The results in Figure 5.7 show that the UGSM model overcomes the mentioned algorithms at all neighbor sizes. Unlike other methods, the UGSM is not much affected when increasing the neighbors' size, as it behaves almost stable with different neighbors' sizes. When k = 50, most of the tested algorithms have improved accuracy. However, the UGSM accuracy is still the best; the prediction accuracy improved by 24% against the UW method, 17% against the PIP method, and more than that against the PCC and COS.

These results reflect the ability of UGSM to choose the appropriate neighbors for the active user based on the global user's taste for the item genre, thus increasing the power of the prediction model based on the local similarity among users.

Type 2: Experiment of a new user with an insufficient number of ratings

According to (Ahn. 2008), the accuracy and quality of recommender systems are decreasing when entity information is insufficient. To check the UGSM accuracy, we checked artificial users by establishing the UGSM to handle only a few ratings from the data sets. Similar to (Jesús Bobadilla et al. 2012), we showed the number of ratings from 2 to 20. Then, we employed UGSM, PCC, and COS similarity measures to produce predicted ratings; after that, we evaluated the result using MAE. Due to the long execution time, we randomly selected 100 users from each data set to execute this experiment. Figures 5.8 (a) and (b) are display MAE for different methods outlined upon a different number of ratings on ML-20M and R2-Yahoo music datasets, respectively.

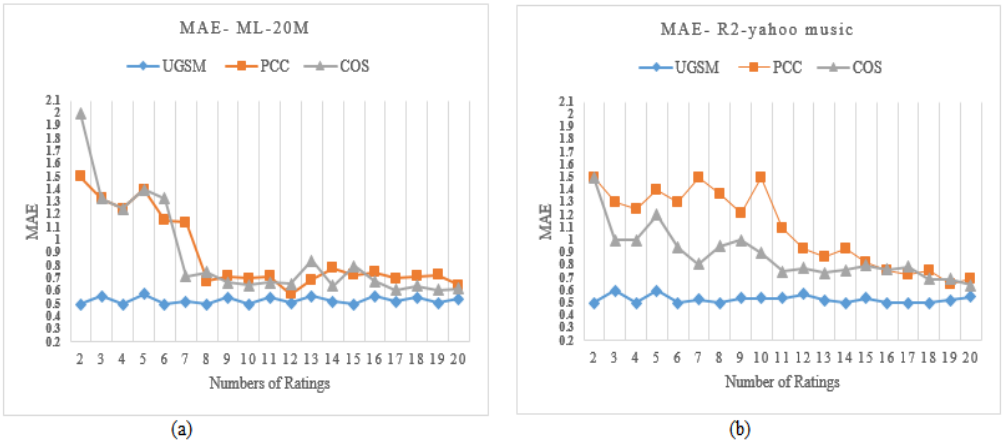

Figure 5.8. *MAE for **(a)** ML-20M data set; **(b)** R2-Yahoo music data set. The X-axis is the number of ratings, and the Y-axis is the MAE value.*

The results demonstrated that the UGSM behaves stably and produced better results than traditional similarity measures in all numbers of ratings for both datasets. UGSM has the successes notably better than PCC and COS even with very few available ratings; like, when the number of ratings =2, 3, or 4.

Type 3: Experiments of pure user cold-start problem

In this experiment, the proposed user cold-start solution in UGSM is evaluated using MAE for prediction accuracy, and compared with (Yadav, Duhan, and Bhatia 2020), which is used the ML-1M data set. During the evaluation, several sizes of neighbor (K-NN) are determined, K from 10 to 100, with increments of 10 per step. Figure 5.9 shows the comparison.

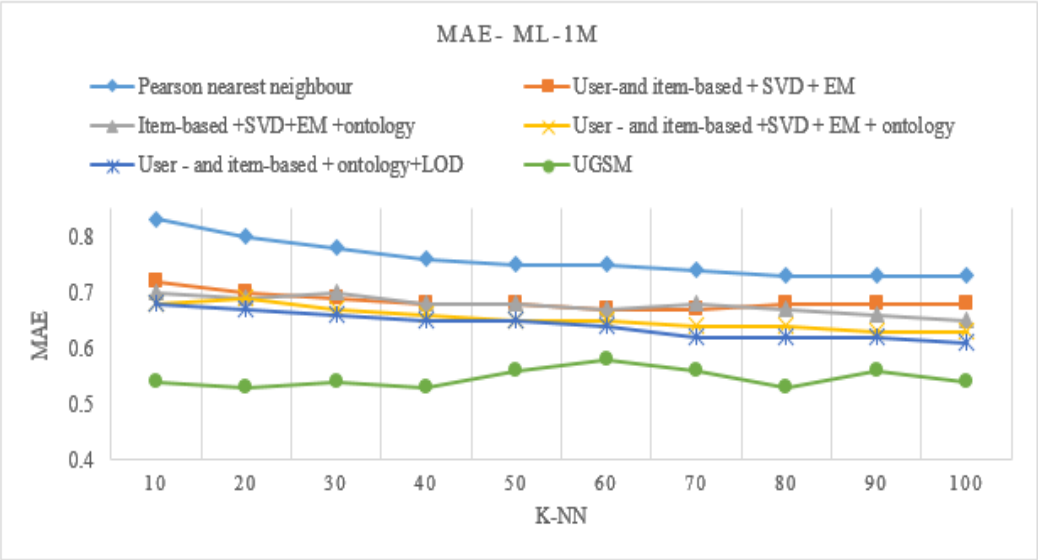

Figure 5.9. *MAE of user cold-start methods using ML-1M data set. The X-axis is the number of ratings, and the Y-axis is the MAE value.*

The results show that the proposed cold-start solution outperforms the tested models in all neighbor's sizes. Furthermore, the prediction accuracy improved by 5.0% against the "User - and item-based + ontology+LOD" method and more than that against the rest tested methods.

5.4. Summary

Online consumers and items have expanded significantly as a result of the proliferation of websites and e-commerce sites. As a competitive enterprise, e-commerce must seek solutions that will increase sales, competitive advantages, and customer happiness. These marketplaces benefit from recommendation systems, which provide customers with recommendations based on their prior choices. As a result, recommendation algorithms are frequently used in conjunction with other domains of knowledge (Linden, Smith, and York 2003).

Because users' moods vary, relying just on user ratings for products may result in erroneous recommendations. As a result, it is critical to introduce a new similarity metric that improves suggestion accuracy even when customers leave few ratings.

This study proposes a novel similarity metric based on item genre information to account for users' preferences and resolve the user cold-start problem. The study's primary contribution is to generate individuals' likenesses using information about their chosen

items' genres, as people generally enjoy specific genres. As a result, two distinct types of user's neighbors have been identified to learn about users' relationships. The first type is the initial neighbors' group, which has users' relationships with each genre separately, who share their preferences with the current user in that genre on an average basis. The second category is the final neighbors' group, consisting of users who share the active user's interests across all genres.

Although the sparsity level varies across the used data sets, the UGSM performs consistently outperforms existing similarity measures with smaller neighborhood sizes in regular user studies and with the cold-start user.

Generally, recommender systems utilize a similarity measure to group users who have cast an adequate number of ratings together and another action to group users who have cast an insufficient number of ratings. However, this study's proposed model (UGSM) provided accurate findings for both user types simultaneously.

6. CONCLUSION AND FUTURE WORK

This chapter includes the conclusions drawn from the proposed models and future work for each solution.

Recommender systems are web tools that assist users in overcoming the overload of information, boost E-Commerce profits, improve sell opportunities, and enhance consumer support. Despite nearly two decades of ever-growing recommender systems use, various novel aspects of them might be investigated.

This dissertation studies focus on managing local and side information of entities to boost similarity measures for the collaborative filtering algorithm and improve prediction accuracy. While local similarity methods rely on co-rated items to determine entities' similarity, side information can aid in identifying similar users regardless of their number of co-rated items.

There are several models available in the RS for determining user or item similarity. However, there are few techniques for determining the connection between the influence of a multi-interesting user and the degree to which the item (user) rating weight is dependent on the user (item) rating.

In Chapter 3, we considered that it is required to carefully consider all relevant information about the system's entities to create a sufficiently reliable recommendation system. Considering that the regression technique can identify which characteristics have a crucial influence on the projected output and establish relationships between distinct variables, we believe that the regression method may quantify the entities' critical impact.

We incorporate key effect differences across connected entities to generate appropriate neighbors for entities. As a result, the solution includes a novel PCC and regression-based neighbor conceptual framework dubbed the PCC-Slope-based model that supports CF. We use a novel prediction model to forecast the entity's rating depending on the first accessible rating from NN and dynamic K-value. The model compared to other novel methods for choosing NNs and to other contemporary CF similarity metrics. Empirical assessments indicate that the proposed model outperforms the other metrics stated previously in terms of accuracy.

As a result, the nearby neighbors must associate with the active user and have a similar influence degree. The findings emphasize the important characteristics of utilizing

entity-level information, such as the degree of relevance to select appropriate neighborhoods and provide a suitable recommendation system. Normally, the neighbors' size affects forecast accuracy, but this approach has demonstrated its accuracy even when the neighbors' sizes are tiny and dynamic.

In the future, we may utilize a mixture model to create a hybrid approach for the user- or item-based RS, which is more successful at predicting the behavior of users with various interests. Additionally, we want to undertake a more comprehensive assessment of the performance using a dense data set. Additional tests may be run to forecast results for various variables, particularly efficiency and diversity.

Another method to discover user similarity is introduced in chapter 4, where the model concentrates only on the common items to infer the users' similarities. The study first investigates the weaknesses of the traditional similarity measures. Then, it introduces a new similarity measure called RDLD (Rating Dissimilarity and largest difference) for collaborative filtering. The RDLD measure uses domain interpretation of users' ratings on goods to conquer the shortcoming of traditional similarities in determining the proper neighborhood and new user cold-start problems. It analyses the difference in users' common rating and the largest weight of these differences.

The experiments are tested on two available data sets to confirm RDLD effectiveness. It presents a high performance for selecting correct neighbors and solving a user cold-start problem in addition to beating some memory-based algorithms and model-based algorithms. It needs no additional information rather than only the user ratings.

The majority of research that addressed the cold-starting issue proposes hybrid approaches that combine content-based and rating data simultaneously, focusing on cold-start situations in which no ratings are accessible. The RDLD approach is focused differently than in prior research; it seeks to enhance existing collaborative filtering methods by developing new user cold-start circumstances with one accessible rating or more.

Future research areas for this model include modifying and using the RDLD similarity measure to various types of product recommendations; for instance, item-based collaborative filtering or clustering approaches may be used to alter and validate the

RDLD measure and its performance. Additionally, examining various performance features of RDLD may be interesting and necessary for determining which factors contribute to RDLD's correct operation.

To improve the quality of suggestions in the sparse data set, we employ item genre as side information to form an approach for determining users' similarities. As a result, in Chapter 5, a novel similarity measure dubbed UGSM (User-Genre Similarity Model) is proposed. UGSM uses item genre information to account for user similarities even when users have no rating (cold-start) or only a few ratings, i.e., 1 to 20 ratings. As users generally like specific genres, the study's principal contribution is to produce users' likenesses using preferred item genre information. Consequently, two classes of user neighbors have been determined to determine users' relationships.

The first class was the initial neighbor group, which contains user relationships according to every genre separately, who share their preferences with active users in a specific genre, based on average user preferences. The second class is the final neighbors' group, which includes users who share their preferences in all genres with the active user.

Furthermore, we suggested two evaluation metrics to calculate the algorithm accuracy: the level of predicted ratings and the users' reliability. We compared the UGSM to two traditional similarity measures PCC and COS, in terms of MAE, level of predicted ratings, and user reliability to confirm UGSM performance. In addition, UGSM was compared with modern user-based collaborative filtering algorithms.

Although the sparsity level was different in both tested datasets, UGSM performed stably and delivered better results than traditional similarity measures with smaller neighborhood sizes in regular user experiments, users with few ratings, and cold-start users.

Generally, in recommender systems, a similarity measure is used among users with a sufficient number of ratings. Another action is used for users who did not cast an adequate number of ratings. In this model, UGSM delivered accurate results for both user types simultaneously.

The model's future research will incorporate item genre into the item-based CF algorithm, improving the item similarity measure and resolving the item cold-start

problem. Additionally, investigating how user attributes may be used to ensure the efficiency of a user-based CF algorithm.

REFERENCES

Adomavicius, G., M. Bamshad, F. Ricci, and A. Tuzhilin. 2015. 'Context-Aware Recommender Systems'. *Recommender Systems Handbook, Second Edition*, 217–253. https://doi.org/10.1007/978-1-4899-7637-6_6.

Adomavicius, Gediminas, and Alexander Tuzhilin. 2005. 'Toward the next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions'. *IEEE Transactions on Knowledge and Data Engineering* 17 (6): 734–49. https://doi.org/10.1109/TKDE.2005.99.

Agarwal, Deepak, and Bee Chung Chen. 2009. 'Regression-Based Latent Factor Models'. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 19–27. https://doi.org/10.1145/1557019.1557029.

Ahn., Hyung Jun. 2008. 'A New Similarity Measure for Collaborative Filtering to Alleviate the New User Cold-Starting Problem'. *Information Sciences* 178 (1): 37–51. https://doi.org/10.1016/j.ins.2007.07.024.

Ahn, Hyung Jun. 2006. 'Utilizing Popularity Characteristics for Product Recommendation'. *International Journal of Electronic Commerce* 11 (2): 59–80. https://doi.org/10.2753/JEC1086-4415110203.

Ai, Jun, Yayun Liu, Zhan Su, Hui Zhang, and Fengyu Zhao. 2019. 'Link Prediction in Recommender Systems Based on Fuzzy Edge Weight Community Detection'. *Epl* 126 (3). https://doi.org/10.1209/0295-5075/126/38003.

Al-safi, Jehan Kadhim Shareef, and Cihan Kaleli. 2021. 'A Correlation and Slope Based Neighbor Selection Model for Recommender Systems'. In *Kumar R., Mishra B.K., Pattnaik P.K. (Eds) Next Generation of Internet of Things. Lecture Notes in Networks and Systems, Vol 201. Springer, Singapore.* https://doi.org/10.1007/978-981-16-0666-3_20.

Al-safi, Jehan, and Cihan Kaleli. 2021. 'A Similarity Measure Model Based on the Dissimilarity Degree between Users'. *Journal of Information and Optimization Sciences, (in Press).*

Al-Safi, Jehan, and Cihan Kaleli. 2021. 'Item Genre - Based Users Similarity Measure for Recommender Systems'. *Appl. Sci.* 11 (6108): 1–20.

https://doi.org/https://doi.org/10.3390/app11136108.

Al-Shamri, Mohammad Yahya H., and Kamal K. Bharadwaj. 2008. 'Fuzzy-Genetic Approach to Recommender Systems Based on a Novel Hybrid User Model'. *Expert Systems with Applications* 35 (3): 1386–99. https://doi.org/10.1016/j.eswa.2007.08.016.

Anand, Deepa, and Kamal K. Bharadwaj. 2011. 'Utilizing Various Sparsity Measures for Enhancing Accuracy of Collaborative Recommender Systems Based on Local and Global Similarities'. *Expert Systems with Applications* 38 (5): 5101–9. https://doi.org/10.1016/j.eswa.2010.09.141.

Arekar, T., M. R. Sonar, and N. J. and Uke. 2015. 'A Survey on Recommendation System'. *International Journal of Innovatice Research in Advanced Engineering* 2 (1): 1–5. http://www.ijirae.com/volumes/Vol2/iss1/01.JACS10081.pdf.

Asanov, Daniar. 2011. 'Algorithms and Methods in Recommender Systems'. *Institute of Technology, Berlin, Germany*. https://doi.org/10.1038/367302a0.

Ayub, Mubbashir, Mustansar Ali Ghazanfar, Tasawer Khan, and Asjad Saleem. 2020. 'An Effective Model for Jaccard Coefficient to Increase the Performance of Collaborative Filtering'. *Arabian Journal for Science and Engineering* 45 (12): 9997–10017. https://doi.org/10.1007/s13369-020-04568-6.

Baltrunas, Linas, and Francesco Ricci. 2008. 'Locally Adaptive Neighborhood Selection for Collaborative Filtering Recommendations'. In *W. Nejdl, J. Kay, P. Pu, E. Herder (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, Vol. 51, Springer, Berlin Heidelberg*, 5149 LNCS:22–31. https://doi.org/10.1007/978-3-540-70987-9_5.

Beel, Joeran, Marcel Genzmehr, Stefan Langer, Andreas Nürnberger, and Bela Gipp. 2013. 'A Comparative Analysis of Offline and Online Evaluations and Discussion of Research Paper Recommender System Evaluation'. *ACM International Conference Proceeding Series*, 7–14. https://doi.org/10.1145/2532508.2532511.

Beel, Joeran, Bela Gipp, Stefan Langer, and Corinna Breitinger. 2016. 'Research-Paper Recommender Systems: A Literature Survey'. *International Journal on Digital Libraries* 17 (4). https://doi.org/10.1007/s00799-015-0156-0.

Bell, Robert M, Yehuda Koren, Chris Volinsky, Park Ave, and Florham Park. 2007. 'Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems'. In *In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 95–104.

Bellogín, Alejandro, and Pablo Castells. 2009. 'Predicting Neighbor Goodness in Collaborative Filtering'. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, LNAI 5822:605–16.

Billsus, D., D. Billsus, M.J. Pazzani, and M.J. Pazzani. 1998. 'Learning Collaborative Information Filters'. *Proceedings of the Fifteenth International Conference on Machine Learning* 54: 47. http://www.aaai.org/Papers/Workshops/1998/WS-98-08/WS98-08-005.pdf.

Bobadilla, J., F. Serradilla, and J. Bernal. 2010. 'A New Collaborative Filtering Metric That Improves the Behavior of Recommender Systems'. *Knowledge-Based Systems* 23 (6): 520–28. https://doi.org/10.1016/j.knosys.2010.03.009.

Bobadilla, Jesús, Fernando Ortega, and Antonio Hernando. 2012. 'A Collaborative Filtering Similarity Measure Based on Singularities'. *Information Processing and Management* 48 (2): 204–17. https://doi.org/10.1016/j.ipm.2011.03.007.

Bobadilla, Jesús, Fernando Ortega, Antonio Hernando, and Jesús Bernal. 2012. 'A Collaborative Filtering Approach to Mitigate the New User Cold Start Problem'. *Knowledge-Based Systems* 26: 225–38. https://doi.org/10.1016/j.knosys.2011.07.021.

Borràs, Joan, Antonio Moreno, and Aida Valls. 2014. 'Intelligent Tourism Recommender Systems: A Survey'. *Expert Systems with Applications* 41 (16): 7370–89. https://doi.org/10.1016/j.eswa.2014.06.007.

Bostandjiev, Svetlin, John O'Donovan, and Tobias Höllerer. 2012. 'Tasteweights: A Visual Interactive Hybrid Recommender System'. *RecSys'12 - Proceedings of the 6th ACM Conference on Recommender Systems*, no. July 2015: 35–42. https://doi.org/10.1145/2365952.2365964.

Breese, John S., David Heckerman, and Carl Kadie. 1998. 'Empirical Analysis of

Predictive Algorithms for Collaborative Filtering'. *Cooper GF and Moral S, Eds., Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98).*, 43–52. http://arxiv.org/abs/1301.7363.

Burke., Robin. 2007. *Hybrid Web Recommender Systems*. *The Adaptive Web*. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-72079-9.

Burke, Robin. 2002. 'Hybrid Recommender Systems: Survey and Experiments'. *User Modeling and User-Adapted Interaction* 12: 1–30. https://doi.org/10.1023/A.

Cacheda, Fidel, Víctor Carneiro, Diego Fernández, and Vreixo Formoso. 2011. 'Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender Systems'. *ACM Transactions on the Web* 5 (1). https://doi.org/10.1145/1921591.1921593.

Campos, Luis M. De, Juan M. Fernández-Luna, Juan F. Huete, and Miguel A. Rueda-Morales. 2010. 'Combining Content-Based and Collaborative Recommendations: A Hybrid Approach Based on Bayesian Networks'. *International Journal of Approximate Reasoning* 51 (7): 785–99. https://doi.org/10.1016/j.ijar.2010.04.001.

Candillier, Laurent, Meyer Frank, and Fran¸coise Fessant. 2008. 'Designing Specific Weighted Similarity Measures to Improve Collaborative Filtering Systems'. *International Conference of Data Mining (ICDM)* 5077 LNAI (July): 242-255. https://doi.org/10.1007/978-3-540-70720-2.

Charu C. Aggarwal. 2016. *Recommender Systems :The Textbook*. NY, USA ISBN. https://doi.org/10.1145/245108.245121.

Chen, Dan Er. 2009. 'The Collaborative Filtering Recommendation Algorithm Based on BP Neural Networks'. *2009 International Symposium on Intelligent Ubiquitous Computing and Education, IUCE 2009* 1: 234–36. https://doi.org/10.1109/IUCE.2009.121.

Chen, Hao, Zhongkun Li, and Wei Hu. 2016. 'An Improved Collaborative Recommendation Algorithm Based on Optimized User Similarity'. *Journal of Supercomputing* 72 (7): 2565–78. https://doi.org/10.1007/s11227-015-1518-5.

Chen, Lei, Yuyu Yuan, Jincui Yang, and Ahmed Zahir. 2021. 'Improving the Prediction

Quality in Memory-Based Collaborative Filtering Using Categorical Features'. *Electronics (Switzerland)* 10 (2): 1–17. https://doi.org/10.3390/electronics10020214.

Chen, Yifan, and Maarten De Rijke. 2018. 'A Collective Variational Autoencoder for Top-n Recommendation with Side Information'. *ACM International Conference Proceeding Series*, 3–9. https://doi.org/10.1145/3270323.3270326.

Choi, Sang Min, Sang Ki Ko, and Yo Sub Han. 2012. 'A Movie Recommendation Algorithm Based on Genre Correlations'. *Expert Systems with Applications* 39 (9): 8079–85. https://doi.org/10.1016/j.eswa.2012.01.132.

Chrsistian Desrosiers, and George Karypis. 2008. 'Technical Report: Solving the Sparsity Problem: Collaborative Filtering via Indirect Similarities'.

Cremonesi, Paolo, Yehuda Koren, and Roberto Turrin. 2010. 'Performance of Recommender Algorithms on Top-N Recommendation Tasks'. *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems*. https://doi.org/10.1145/1864708.1864721.

Das, Joydeep, Subhashis Majumder, and Kalyani Mali. 2017. 'Context Aware Scalable Collaborative Filtering'. *Proceedings of the 2017 International Conference On Big Data Analytics and Computational Intelligence, ICBDACI 2017*, no. June 2019: 184–90. https://doi.org/10.1109/ICBDACI.2017.8070833.

Dell'Amico, Matteo, and Licia Capra. 2008. 'SOFIA: Social Filtering for Robust Recommendations'. *IFIP International Federation for Information Processing* 263: 135–50. https://doi.org/10.1007/978-0-387-09428-1_9.

Demissie, Solomon. 2018. 'A Clustering-Based Context-Aware Recommender Systems Through Extraction of Latent Preferences'. *International Journal of Advanced Research in Computer Science* 9 (2): 699–707. https://doi.org/10.26483/ijarcs.v9i2.5862.

Deshpande, Mukund, and George Karypis. 2004. 'Item-Based Top-N Recommendation Algorithms'. *ACM Trans. Inf. Syst. 22 (1)*, 1–26.

Frémal, Sébastien, and Fabian Lecron. 2017. 'Weighting Strategies for a Recommender

System Using Item Clustering Based on Genres'. *Expert Systems with Applications* 77: 105–13. https://doi.org/10.1016/j.eswa.2017.01.031.

G, Adomavicius, and Tuzhilin A. 2005. 'Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions'. *IEEE Transactions on Knowledge and Data Engineering* 17 (6): 734–49.

Gantner, Zeno, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. 'Learning Attribute-to-Feature Mappings for Cold-Start Recommendations'. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, 176–85. IEEE. https://doi.org/10.1109/ICDM.2010.129.

Gao, Jingyue, Xiting Wang, Yasha Wang, and Xing Xie. 2019. 'Explainable Recommendation through Attentive Multi-View Learning'. In *33rd AAAI -I9*, 3622–29. https://doi.org/10.1609/aaai.v33i01.33013622.

Garcin, Florent, Boi Faltings, Radu Jurca, and Nadine Joswig. 2009. 'Rating Aggregation in Collaborative Filtering Systems'. *RecSys'09 - Proceedings of the 3rd ACM Conference on Recommender Systems*, 349–52. https://doi.org/10.1145/1639714.1639785.

Gazdar, Achraf, and Lotfi Hidri. 2019. 'A New Similarity Measure for Collaborative Filtering Based Recommender Systems'. *Knowledge-Based Systems* 188 (September). https://doi.org/10.1016/j.knosys.2019.105058.

Gediminas, Adomavicius, Nikos Manouselis, and YoungOk Kwon. 2011. 'Multi-Criteria Recommender Systems'. In *Recommender Systems Handbook*, 769–803. https://doi.org/10.1007/978-0-387-85820-3.

Ghauth, Khairil Imran Bin, and Nor Aniza Abdullah. 2009. 'Building an E-Learning Recommender System Using Vector Space Model and Good Learners Average Rating'. In *2009 9th IEEE International Conference on Advanced Learning Technologies, ICALT*, 194–96. https://doi.org/10.1109/ICALT.2009.161.

Gogna, Anupriya, and Angshul Majumdar. 2015. 'A Comprehensive Recommender System Model: Improving Accuracy for Both Warm and Cold Start Users'. *IEEE Access* 3: 2803–13. https://doi.org/10.1109/ACCESS.2015.2510659.

Goldberg, David, David Nichols, Brian M. Oki, and Douglas Terry. 1992. 'Using Collaborative Filtering to Weave an Information Tapestry'. *Communications of the ACM* 35 (12): 61–70. https://doi.org/10.1145/138859.138867.

Guo, Qingyu, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. 'A Survey on Knowledge Graph-Based Recommender Systems'. *IEEE Transactions on Knowledge and Data Engineering* XX (XX): 1–20. https://doi.org/10.1109/tkde.2020.3028705.

Han, Xiaotian, Chuan Shi, Senzhang Wang, Philip S. Yu, and Li Song. 2018. 'Aspect-Level Deep Collaborative Filtering via Heterogeneous Information Networks'. *IJCAI*, 3393–99. https://doi.org/10.24963/ijcai.2018/471.

Harper, F. Maxwell, and Joseph A. Konstan. 2015. 'The Movielens Datasets: History and Context'. *ACM Transactions on Interactive Intelligent Systems* 5 (4): 1–20. https://doi.org/0000001.0000001.

Hasan, Mahamudul, and Falguni Roy. 2019. 'An Item–Item Collaborative Filtering Recommender System Using Trust and Genre to Address the Cold-Start Problem'. *Big Data and Cognitive Computing* 3 (3): 1–15. https://doi.org/10.3390/bdcc3030039.

HAYES, ADAM. 2020. 'R-Squared Definition'. 2020.

He, Xing Sheng, Ming Yang Zhou, Zhao Zhuo, Zhong Qian Fu, and Jian Guo Liu. 2015. 'Predicting Online Ratings Based on the Opinion Spreading Process'. *Physica A: Statistical Mechanics and Its Applications* 436: 658–64. https://doi.org/10.1016/j.physa.2015.05.066.

He, Xuansen, and Xu Jin. 2019. 'Collaborative Filtering Recommendation Algorithm Considering Users' Preferences for Item Attributes'. *Proceedings of the 2019 International Conference on Big Data and Computational Intelligence, ICBDCI 2019*, no. February 2019: 1–6. https://doi.org/10.1109/ICBDCI.2019.8686102.

Herlocker, J. L., J. A. Konstan, A. Borchers, and and J. Riedl. 1999. 'An Algorithmic Framework for Performing Collaborative Filtering'. *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* 51: 227–34.

Herlocker, Jon, Joseph A. Konstan, and John Riedl. 2002. 'An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms'. *Information Retrieval* 5 (4): 287–310. https://doi.org/10.1023/A:1020443909834.

Hu, Liang, Songlei Jian, Longbing Cao, Zhiping Gu, Qingkui Chen, and Artak Amirbekyan. 2019. 'HERS: Modeling Influential Contexts with Heterogeneous Relations for Sparse and Cold-Start Recommendation'. In *33 AAAI*, 2:3830–37. https://doi.org/10.1609/aaai.v33i01.33013830.

Huang, Meigen, Yu Wang, and Lihan Zhou. 2019. 'Collaborative Filtering Algorithm Based on Linear Regression Filling'. *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019*, no. Itnec: 1831–34. https://doi.org/10.1109/ITNEC.2019.8728971.

Huang, Zan, Hsinchun Chen, and Daniel Zeng. 2004. 'Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering'. *ACM Transactions on Information Systems* 22 (1): 116–42. https://doi.org/10.1145/963770.963775.

'IMDb'. 2019. 2019. https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvSU1EYg.

Isinkaye, F. O., Y. O. Folajimi, and B. A. Ojokoh. 2015. 'Recommendation Systems: Principles, Methods and Evaluation'. *Egyptian Informatics Journal* 16 (3): 261–73. https://doi.org/10.1016/j.eij.2015.06.005.

Jain, Sarika, Anjali Grover, Praveen Singh Thakur, and Sourabh Kumar Choudhary. 2015. 'Trends, Problems and Solutions of Recommender System'. *International Conference on Computing, Communication and Automation, ICCCA 2015*, 955–58. https://doi.org/10.1109/CCAA.2015.7148534.

Jiang, Liaoliang, Yuting Cheng, Li Yang, Jing Li, Hongyang Yan, and Xiaoqin Wang. 2018. 'A Trust-Based Collaborative Filtering Algorithm for E-Commerce Recommendation System'. *Journal of Ambient Intelligence and Humanized Computing* 10 (8): 3023–34. https://doi.org/10.1007/s12652-018-0928-7.

Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong.

2019. ‘Taxonomy-Aware Multi-Hop Reasoning Networks for Sequential Recommendation’. *WSDM`19*, 573–581.

Jin, Rong, Joyce Y. Chai, and Luo Si. 2004. ‘An Automatic Weighting Scheme for Collaborative Filtering’. *Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval - SIGIR ’04*, 337. https://doi.org/10.1145/1008992.1009051.

Kim, Heung Nam, Ae Ttie Ji, Inay Ha, and Geun Sik Jo. 2010. ‘Collaborative Filtering Based on Collaborative Tagging for Enhancing the Quality of Recommendation’. *Electronic Commerce Research and Applications* 9 (1): 73–83. https://doi.org/10.1016/j.elerap.2009.08.004.

Kim, Kyoung jae, and Hyunchul Ahn. 2008. ‘A Recommender System Using GA K-Means Clustering in an Online Shopping Market’. *Expert Systems with Applications* 34 (2): 1200–1209. https://doi.org/10.1016/j.eswa.2006.12.025.

Kim, Kyung Rog, and Nammee Moon. 2012. ‘Recommender System Design Using Movie Genre Similarity and Preferred Genres in SmartPhone’. *Multimedia Tools and Applications* 61 (1): 87–104. https://doi.org/10.1007/s11042-011-0728-y.

Kim, Tack Hun, and Sung Bong Yang. 2007. ‘An Effective Threshold-Based Neighbor Selection in Collaborative Filtering’. In *Proceedings of the 29th European Conference on IR Research. ECIR’07, Springer-Verlag, Berlin, Heidelberg*, 4425 LNCS:712–15. https://doi.org/10.1007/978-3-540-71496-5_75.

Koren, Yehuda. 2010. ‘Factor in the Neighbors : Scalable and Accurate Collaborative Filtering’. *ACM Transactions on Knowledge Discovery from Data* 4 (1): 1–24. https://doi.org/10.1145/1644873.1644874.

Koren, Yehuda, Robert Bell, and Chris Volinsky. 2009. ‘Matrix Factorization Techniques for Recommender Systems’. *Computer*, 30–37. https://doi.org/10.1109/MC.2009.263.

Kortenhof, Bas Lennart Van. 2017. ‘Context-Aware Recommender Systems in the E-Commerce Domain’.

Koutrika, Georgia. 2009. ‘FlexRecs : Expressing and Combining Flexible

Recommendations'. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 745–58. https://doi.org/10.1145/1559845.1559923.

Krulwich, Bruce. 1997. 'Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data'. *AI Magazine* 18 (2): 37–45.

Lathia, Neal, Stephen Hailes, and Licia Capra. 2008. 'The Effect of Correlation Coefficients on Communities of Recommenders'. *Proceedings of the ACM Symposium on Applied Computing*, 2000–2005. https://doi.org/10.1145/1363686.1364172.

Lemire, Daniel, and Anna Maclachlan. 2005. 'Slope One Predictors for Online Rating-Based Collaborative Filtering'. *Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005*, 471–75. https://doi.org/10.1137/1.9781611972757.43.

Li, Chaozhuo, Wang Fang, Yang Yang, and Xiaoming Zhang. 2015. 'Exploring Social Network Information for Solving Cold Start in Product Recommendation'. In *Conference: International Conference on Web Information Systems Engineering*. Vol. 9419. https://doi.org/DOI: 10.1007/978-3-319-26187-4_24.

Li, Jing, Wentao Xu, Wenbo Wan, and Jiande Sun. 2018. 'Movie Recommendation Based on Bridging Movie Feature and User Interest'. *Journal of Computational Science* 26: 128–34. https://doi.org/10.1016/j.jocs.2018.03.009.

Li, Qing, and Byeong Man Kim. 2003. 'Clustering Approach for Hybrid Recommender System'. *Proceedings - IEEE/WIC International Conference on Web Intelligence, WI 2003*, 33–38. https://doi.org/10.1109/WI.2003.1241167.

Li, Shangsong, and Xuesong Li. 2020. 'Collaborative Filtering Recommendation Algorithm Based on User Characteristics and User Interests'. *Journal of Physics: Conference Series* 1616 (1): 1–11. https://doi.org/10.1088/1742-6596/1616/1/012032.

Liang, Zhang, Xiao Bo, and Guo Jun. 2009. 'An Approach of Selecting Right Neighbors for Collaborative Filtering'. *Proceedings of the Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference*, 1057–60. https://doi.org/10.1109/ICICIC.2009.76.

Linden, Greg, Brent Smith, and Jeremy York. 2003. 'Amazon.Com Recommendations: Item-to-Item Collaborative Filtering'. *Internet Comput IEEE*, no. February: 76–80. https://doi.org/10.1109/MIC.2003.1167344.

Liu, Chun Yi, Chuan Zhou, Jia Wu, Yue Hu, and Li Guo. 2018. 'Social Recommendation with an Essential Preference Space'. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 346–53.

Liu, Haifeng, Zheng Hu, Ahmad Mian, Hui Tian, and Xuzhen Zhu. 2014. 'A New User Similarity Model to Improve the Accuracy of Collaborative Filtering'. *Knowledge-Based Systems* 56: 156–66. https://doi.org/10.1016/j.knosys.2013.11.006.

Liu, Run Ran, Chun Xiao Jia, Tao Zhou, Duo Sun, and Bing Hong Wang. 2009. 'Personal Recommendation via Modified Collaborative Filtering'. *Physica A: Statistical Mechanics and Its Applications* 388 (4): 462–68. https://doi.org/10.1016/j.physa.2008.10.010.

Lu, Jie, Qusai Shambour, Yisi Xu, Qing Lin, and Guangquan Zhang. 2010. 'BizSeeker: A Hybrid Semantic Recommendation System for Personalized Government-to-Business e-Services'. *Internet Research* 20 (3): 342–65. https://doi.org/10.1108/10662241011050740.

Lu, Zhigang, and Hong Shen. 2015. 'A Security-Assured Accuracy-Maximised Privacy Preserving Collaborative Filtering Recommendation Algorithm'. *ACM International Conference Proceeding Series* 0 (CONFCODENUMBER): 1–20. https://doi.org/10.1145/2790755.2790757.

Luo, Heng, Changyong Niu, Ruimin Shen, and Carsten Ullrich. 2008. 'A Collaborative Filtering Framework Based on Both Local User Similarity and Global User Similarity'. *Machine Learning* 72 (3): 231–45. https://doi.org/10.1007/s10994-008-5068-4.

M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. 1999. 'Combining Content-Based and Collaborative Filters in an Online Newspaper'. In *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation. Berkeley, California*.

Middleton, Stuart E., Harith Alani, and David C. De Roure. 2002. 'Exploiting Synergy

Between Ontologies and Recommender Systems'. *Semantic Web Workshop* 55: 41–50. http://uk.arxiv.org/abs/cs/0204012v1.

Moscato, Vincenzo, Antonio Picariello, and Giancarlo Sperli. 2020. 'An Emotional Recommender System for Music'. *IEEE Intelligent Systems* XX (XX): 1–10. https://doi.org/10.1109/MIS.2020.3026000.

Nadimi-Shahraki, Mohammad Hossein, and Mozhde Bahadorpour. 2014. 'Cold-Start Problem in Collaborative Recommender Systems: Efficient Methods Based on Ask-to-Rate Technique'. *Journal of Computing and Information Technology* 22 (2): 105–13. https://doi.org/10.2498/cit.1002223.

O'Connor, M, and J Herlocker. 2001. 'Clustering Items for Collaborative Filtering'. *Human Factors*, no. June. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.6325&rep=rep1&type=pdf.

O'donovan, John, Brynjar Gretarsson, Svetlin Bostandjiev, Tobias Höllerer, and Barry Smyth. 2009. 'A Visual Interface for Social Information Filtering'. *Proceedings - 12th IEEE International Conference on Computational Science and Engineering, CSE 2009* 4 (October): 74–81. https://doi.org/10.1109/CSE.2009.26.

O'Donovan, John, Barry Smyth, Brynjar Gretarsson, Svetlin Bostandjiev, and Tobias Höllerer. 2008. 'PeerChooser: Visual Interactive Recommendation'. *Conference on Human Factors in Computing Systems - Proceedings*, no. July 2015: 1085–88. https://doi.org/10.1145/1357054.1357222.

Paradarami, Tulasi K., Nathaniel D. Bastian, and Jennifer L. Wightman. 2017. 'A Hybrid Recommender System Using Artificial Neural Networks'. *Expert Systems with Applications* 83: 300–313. https://doi.org/10.1016/j.eswa.2017.04.046.

Park, Seung Taek, David Pennock, Omid Madani, Nathan Good, and Dennis DeCoste. 2006. 'Naïve Filterbots for Robust Cold-Start Recommendations'. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 699–705. https://doi.org/10.1145/1150402.1150490.

Park, Youngki, Sungchan Park, Woosung Jung, and Sang Goo Lee. 2015. 'Reversed CF: A Fast Collaborative Filtering Algorithm Using a k-Nearest Neighbor Graph'.

Expert Systems with Applications 42 (8): 4022–28. https://doi.org/10.1016/j.eswa.2015.01.001.

Pazzani, Michael J. 1999. 'Framework for Collaborative, Content-Based and Demographic Filtering'. *Artificial Intelligence Review* 13 (5): 393–408. https://doi.org/10.1023/a:1006544522159.

Pazzani, Michael J, and Daniel Billsus. 2007. 'Content-Based Recommendation Systems'. In *The Adaptive Web*, 325–41.

Qian, Xueming, He Feng, Guoshuai Zhao, and Tao Mei. 2014. 'Personalized Recommendation Combining User Interest and Social Circle'. *IEEE Transactions on Knowledge and Data Engineering* 26 (7): 1763–77. https://doi.org/10.1109/TKDE.2013.168.

Quadrana, Massimo, Dietmar Jannach, and Paolo Cremonesi. 2018. 'Sequence-Aware Recommender Systems'. *ACM Comput. Surv.* 51 (4): 1–36. https://doi.org/10.1145/3308560.3320091.

Rashid, Al Mamunur, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, and John Riedl. 2002. 'Getting to Know You: Learning New User Preferences in Recommender Systems'. *International Conference on Intelligent User Interfaces, Proceedings IUI*, 127–34.

Ren, Lei, Liang He, Junzhong Gu, Weiwei Xia, and Faqing Wu. 2008. 'A Hybrid Recommender Approach Based on Widrow-Hoff Learning'. *Proceedings of the 2008 2nd International Conference on Future Generation Communication and Networking, FGCN 2008* 1: 40–45. https://doi.org/10.1109/FGCN.2008.48.

Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. 'GroupLens: An Open Architecture for Collaborative Filtering of Netnews'. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW 1994*, 175–86. https://doi.org/10.1145/192844.192905.

Ricci, Francesco, Lior Rokach, and Bracha Shapira. 2014. 'Introduction to Recommender Systems Handbook'. In *Recommender Systems Handbook*, 1–39. https://doi.org/10.1007/978-0-387-85820-3.

Ricci, Francesco, Bracha Shapira, and Lior Rokach. 2015. *Recommender Systems Handbook, Second Edition*. *Recommender Systems Handbook, Second Edition*. https://doi.org/10.1007/978-1-4899-7637-6.

Rosli, Ahmad Nurzid, Tithrottanak You, Inay Ha, Kyung Yong Chung, and Geun Sik Jo. 2015. 'Alleviating the Cold-Start Problem by Incorporating Movies Facebook Pages'. *Cluster Computing* 18 (1): 187–97. https://doi.org/10.1007/s10586-014-0355-2.

Santos, Edson B., Rudinei Goularte, and Marcelo G. Manzato. 2014. 'Personalized Collaborative Filtering: A Neighborhood Model Based on Contextual Constraints'. *Proceedings of the ACM Symposium on Applied Computing*, 919–24. https://doi.org/10.1145/2554850.2555017.

Sarstedt, Marko, and Erik Mooi. 2014. *A Concise Guide to Market Research - Chapter 7 Regression Analysis*. *Springer Texts in Business and Economics*. https://doi.org/10.1007/978-3-642-53965-7.

Sarwar, B., George Karypis, Joseph Konstan, and John Riedl. 2001. 'Item-Based Collaborative Filtering Recommendation Algorithms'. *Proceedings of the 10th International Conference on World Wide Web, WWW 2001*, no. August: 285–95. https://doi.org/10.1145/371920.372071.

Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. 2000. 'Analysis of Recommendation Algorithms for E-Commerce', 158–67. https://doi.org/10.1145/352871.352887.

Schafer, Ben J., Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. 'Collaborative Filtering Recommender Systems'. *TheAdaptiveWeb,Springer,2007,* 4321: 291–324. http://www.eui.upm.es/~jbobi/jbobi/PapersRS/CollaborativeFilteringRecommenderSystems.pdf.

Schafer, J. Ben, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. 'Collaborative Filtering Recommender Systems'. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4321 LNCS (1): 291–324. https://doi.org/10.1007/978-3-540-72079-9_9.

Shardanand, Upendra, and Pattie Maes. 1995. ‘Social Information Filtering: Algorithms for Automating “Word of Mouth”’’’’. *ENR (Engineering News-Record)*, 1–15.

Sielis, George A., Aimilia Tzanavari, and George A. Papadopoulos. 2014. ‘Recommender Systems Review of Types, Techniques, and Applications’. *Encyclopedia of Information Science and Technology, Third Edition*, 7260–70. https://doi.org/10.4018/978-1-4666-5888-2.ch714.

Singh, Ajit P., and Geoffrey J. Gordon. 2008. ‘Relational Learning via Collective Matrix Factorization’. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 650–58. https://doi.org/10.1145/1401890.1401969.

Soares, Márcio, and Paula Viana. 2015. ‘Tuning Metadata for Better Movie Content-Based Recommendation Systems’. *Multimedia Tools and Applications* 74 (17): 7015–36. https://doi.org/10.1007/s11042-014-1950-1.

Soojung Lee. 2017. ‘Improving Jaccard Index for Measuring Similarity in Collaborative Filtering’. In *Lecture Notes in Electrical Engineering*, 424:799–806. https://doi.org/10.1007/978-981-10-4154-9_71.

Stephanie. 2016. ‘Statistics How To RMSE : Root Mean Square Error What Is Root Mean Square Error’. 2016. https://www.statisticshowto.datasciencecentral.com/rmse/.

Su, Xiaoyuan, and Taghi M. Khoshgoftaar. 2009. ‘A Survey of Collaborative Filtering Techniques’. *Advances in Artificial Intelligence* 2009: 1–19. https://doi.org/10.1155/2009/421425.

Su, Zhan, Xiliang Zheng, Jun Ai, Yuming Shen, and Xuanxiong Zhang. 2020. ‘Link Prediction in Recommender Systems Based on Vector Similarity’. *Physica A: Statistical Mechanics and Its Applications* 560: 125154. https://doi.org/10.1016/j.physa.2020.125154.

Subramaniyaswamy, V., and R. Logesh. 2017. ‘Adaptive KNN Based Recommender System through Mining of User Preferences’. *Wireless Personal Communications* 97 (2): 2229–47. https://doi.org/10.1007/s11277-017-4605-5.

Sun, Hui-Feng, Jun-Liang Chen, Gang Yu, Chuan-Chang Liu, Yong Peng, Guang Chen,

and Bo Cheng. 2012. 'JacUOD: A New Similarity Measurement for Collaborative Filtering'. *Journal of Computer Science and Technology* 27 (6): 1252–60. https://doi.org/10.1007/s11390-012-1301-5.

'Surprise Prediction_algorithms Package'. 2020. 2020. https://surprise.readthedocs.io/en/stable/prediction_algorithms_package.html.

Ur Rehman, Zia, Farookh K. Hussain, and Omar K. Hussain. 2013. 'Frequency-Based Similarity Measure for Multimedia Recommender Systems'. *Multimedia Systems* 19 (2): 95–102. https://doi.org/10.1007/s00530-012-0281-1.

Veras De Sena Rosa, Ricardo Erikson, Felipe Augusto Souza Guimaraes, Rafael da Silva Mendonca, and Vicente Ferreira de Lucena. 2020. 'Improving Prediction Accuracy in Neighborhood-Based Collaborative Filtering by Using Local Similarity'. *IEEE Access* 8: 142795–809. https://doi.org/10.1109/access.2020.3013733.

Villegas, Norha M., Cristian Sánchez, Javier Díaz-Cely, and Gabriel Tamura. 2018. 'Characterizing Context-Aware Recommender Systems: A Systematic Literature Review'. *Knowledge-Based Systems* 140: 173–200. https://doi.org/10.1016/j.knosys.2017.11.003.

Vincent Schickel-Zuber, Boi Faltings. 2005. 'Heterogeneous Attribute Utility Model: A New Approach for Modeling User Profiles for Recommendation Systems'. In *August 21, 2005 Chicago, Illinois, USA In Conjunction with The 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2005)*, 19:80–91. https://doi.org/10.1039/C7CP03578K.

Wang, Dawei, Yuehwern Yih, and Mario Ventresca. 2020. 'Improving Neighbor-Based Collaborative Filtering by Using a Hybrid Similarity Measurement'. *Expert Systems with Applications* 160: 1–17. https://doi.org/10.1016/j.eswa.2020.113651.

Wang, Hongwei, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. 'RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems'. *International Conference on Information and Knowledge Management, Proceedings*, no. 1: 417–26. https://doi.org/10.1145/3269206.3271739.

Wang, Shoujin, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet

Orgun. 2019. ‘Sequential Recommender Systems: Challenges, Progress and Prospects’. *IJCAI*, 6332–38. https://doi.org/10.24963/ijcai.2019/883.

Wang, Wen, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. ‘Beyond Clicks: Modeling Multi-Relational Item Graph for Session-Based Target Behavior Prediction’. *The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020*, 3056–3062. https://doi.org/10.1145/3366423.3380077.

Wang, Yong, Jiangzhou Deng, Jerry Gao, and Pu Zhang. 2017. ‘A Hybrid User Similarity Model for Collaborative Filtering’. *Information Sciences* 418–419. https://doi.org/10.1016/j.ins.2017.08.008.

Willmott, Cort J., and Kenji Matsuura. 2005. ‘Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance’. *Climate Research* 30 (1): 79–82. https://doi.org/10.3354/cr030079.

Xiao, Jun, Minjuan Wang, Bingqian Jiang, and Junli Li. 2018. ‘A Personalized Recommendation System with Combinational Algorithm for Online Learning’. *Journal of Ambient Intelligence and Humanized Computing* 9 (3): 667–77. https://doi.org/10.1007/s12652-017-0466-8.

Xue, Gui Rong, Chenxi Lin, Qiang Yang, Wensi Xi, Hua Jun Zeng, Yong Yu, and Zheng Chen. 2005. ‘Scalable Collaborative Filtering Using Cluster-Based Smoothing’. *SIGIR 2005 - Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 114–21. https://doi.org/10.1145/1076034.1076056.

Yadav, Usha, Neelam Duhan, and Komal Kumar Bhatia. 2020. ‘Dealing with Pure New User Cold-Start Problem in Recommendation System Based on Linked Open Data and Social Network Features’. *Mobile Information Systems* 2020: 1–20. https://doi.org/10.1155/2020/8912065.

Yang, Zhe, Bing Wu, Kan Zheng, Xianbin Wang, and Lei Lei. 2016. ‘A Survey of Collaborative Filtering-Based Recommender Systems for Mobile Internet Applications’. *IEEE Access* 4: 3273–87. https://doi.org/10.1109/ACCESS.2016.2573314.

You, Tithrottanak, Ahmad Nurzid Rosli, Inay Ha, and Geun-Sik Jo. 2013. 'Clustering Method Based on Genre Interest for Cold-Start Problem in Movie Recommendation'. *Journal of Intelligence and Information Systems* 19 (1): 57–77. https://doi.org/10.13088/jiis.2013.19.1.057.

Yuan, Yuyu, Ahmed Zahir, and Jincui Yang. 2019. 'Modeling Implicit Trust in Matrix Factorization-Based Collaborative Filtering'. *Applied Sciences (Switzerland)* 9 (20). https://doi.org/10.3390/app9204378.

Yue Shi, Martha Larson, and Alan Hanjalic. 2014. 'Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges'. *ACM Computing Surveys (CSUR)* 47 (1): 1–45.

Zare, Hadi, Nikooie Pour, Abd Mina, and Parham Moradi. 2019. 'Enhanced Recommender System Using Predictive Network Approach'. *Physica A: Statistical Mechanics and Its Applications* 520: 322–37. https://doi.org/10.1016/j.physa.2019.01.053.

Zeng, Chun, Chun Xiao Xing, Li Zhu Zhou, and Xiao Hui Zheng. 2004. 'Similarity Measure and Instance Selection for Collaborative Filtering'. *International Journal of Electronic Commerce* 8 (4): 115–29. https://doi.org/10.1080/10864415.2004.11044314.

Zeybek, Halil, and Cihan KALELİ. 2018. 'Dynamic k Neighbor Selection for Collaborative Filtering'. *ANADOLU UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY A - Applied Sciences and Engineering*, 1–13. https://doi.org/10.18038/aubtda.346407.

Zhang, Heng Ru, Fan Min, and Bing Shi. 2017. 'Regression-Based Three-Way Recommendation'. *Information Sciences* 378: 444–61. https://doi.org/10.1016/j.ins.2016.03.019.

Zhang, Muhan, and Yixin Chen. 2020. 'Inductive Matrix Completion Based on Graph Neural Networks'. *ICLR*, no. Imc: 1–14. http://arxiv.org/abs/1904.12058v3.

Zhang, Shuai, Lina Yao, Aixin Sun, and Yi Tay. 2019. 'Deep Learning Based Recommender System: A Survey and New Perspectives'. *ACM Computing Surveys* 52 (1): 1–35. https://doi.org/10.1145/3285029.

Zhang, Ye, and Wei Song. 2009. ‘A Collaborative Filtering Recommendation Algorithm Based on Item Genre and Rating Similarity’. *Proceedings of the 2009 International Conference on Computational Intelligence and Natural Computing, CINC 2009*, no. 2: 72–75. https://doi.org/10.1109/CINC.2009.219.

Zhu, Bo, Remigio Hurtado, Jesus Bobadilla, and Fernando Ortega. 2018. ‘An Efficient Recommender System Method Based on the Numerical Relevances and the Non-Numerical Structures of the Ratings’. *IEEE Access* 6: 49935–54. https://doi.org/10.1109/ACCESS.2018.2868464.

CURRICULUM VITAE

- **Personal Information**

Name Jehan Kadhim Shareef AL-SAFI

- **Languages**

Arabic	Mother language
English	Fluent Level
Turkish	Fluent Level

- **Publications during the PhD. study:**

1- Al-Safi, Jehan Kadhim Shareef, and Cihan Kaleli. 2021. 'A Correlation and Slope Based Neighbor Selection Model for Recommender Systems'. In *Kumar R., Mishra B.K., Pattnaik P.K. (Eds) Next Generation of Internet of Things. Lecture Notes in Networks and Systems, Vol 201. Springer, Singapore.* https://doi.org/10.1007/978-981-16-0666-3_20.

2- Al-Safi, Jehan, and Cihan Kaleli. 2021. 'A Similarity Measure Model Based on the Dissimilarity Degree between Users'. *Journal of Information and Optimization Sciences, (in Press).*

3- Al-Safi, Jehan, and Cihan Kaleli. 2021. 'Item Genre - Based Users Similarity Measure for Recommender Systems'. *Appl. Sci.* 11 (6108): 1–20. https://doi.org/https://doi.org/10.3390/app11136108.