CROSS-DOMAIN RECOMMENDATION SYSTEM FOR IMPROVING ACCURACY BY FOCUSING ON DIVERSITY

Athauda Arachchilage Heshan Wikasitha Sandakelum Herath

209330K

Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2022

CROSS-DOMAIN RECOMMENDATION SYSTEM FOR IMPROVING ACCURACY BY FOCUSING ON DIVERSITY

Athauda Arachchilage Heshan Wikasitha Sandakelum Herath

209330K

Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2022

Declaration

I declare that this is my own work, and this dissertation does not incorporate without acknowledgment any material previously submitted for a degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other media. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: …………………… Date:

Name: A.A.H.W.S. Herath

The above candidate has carried out research for the master's thesis/dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of the supervisor: Dr Sapumal Ahangama

Signature of the supervisor: ……………………. Date: …………………

Acknowledgement

I would like to express profound gratitude to my advisor, Dr. Sapumal Ahangama, for his invaluable support by providing relevant knowledge, materials, advice, supervision, and useful suggestions throughout this research work. His expertise and continuous guidance enabled me to complete my work successfully.

Furthermore, I would like to thank all my colleagues for their help in finding relevant research material, for sharing knowledge and experience, and for their encouragement. I am as ever, especially indebted to my parents and my brother for their love and support throughout my life. Finally, I wish to express my gratitude to all my colleagues, for the support given to me to manage my MSc research work.

.

Abstract

With the rapidly developing technology world, recommender systems also improving day by day since customer expectations also vary from new angles making new business trends. As a result of this kind of situation, enterprise-level recommender systems require more modifications with new improvements to achieve a high user satisfaction level. in that case it seems currently most commercial recommender systems are struggling with low recommender quality by decreasing user trust and expectations. On the other hand, it senses only the recommender accuracy is not sufficient to measure recommender quality. Under the major domain recommender system, the cross-domain recommender system is one of the not much-explored areas and it needs more research works focused on diversity like subjective metrics rather than accuracy. With the purpose of improving accuracy by focusing diversity on CDRS here, I have built a matrix factorization-based collaborative filtering cross-domain recommender system using explicit user feedback with movilens100k research data set. When it comes to cross-domain recommender systems, the most frequent approach is to measure and evaluate their relevancy using standard predicted accuracy metrics such as root mean squared error (RMSE), mean absolute error (MAE), and so on. Since the more need than accuracy to maintain high-quality recommendations, we need to pay attention to a few specific areas beyond accuracy like diversity and novelty. We have measured our CDRS model's performance via RMSE, MSE, MAE, FCP, hit ratio, and Precision@k and in all cases, CDRS has achieved good performance than the general CF model. Moreover, we measured the CDRS model's diversity and novelty and could see both are increasing when top-N increasing. These findings would be pretty much worthy when we are implementing enterprise-level cross-domain recommender systems in the future to achieve success in each modern business use case with enhancing user satisfaction.

TABLE OF CONTENT

LIST OF FIGURES

Figure Description Page

LIST OF TABLES

Table Description Page

LIST OF ABBREVIATIONS

Abbreviation	Description
CBRS	Content-Based Recommender System
CDRS	Cross Domain Recommender System
CF	Collaborative Filtering
CFRS	Collaborative Filtering Recommender System
DRS	Demographic Recommendation System
D_S	Source Domain
D_T	Target Domain
FM	Factorization Machines
HRS	Hybrid Recommender System
KBRS	Knowledge-Based Recommender System
MF	Matrix Factorization
RS	Recommender System

1: INTRODUCTION

The first chapter of the report contains the report's introduction. Section 1.1 provides an overview of the context in which this research was carried out. The introduction of recommender systems is covered in Section 1.2.

1.1: Background

With the rapidly developing technology, the modern world is just a global village, and everyone is connected together via the internet and day-to-day life is now pretty easy since all over services are accessible via the internet quickly. Everything, however, has a flip side. Human thinking skills have become ever more burdened by the rising complexity and heterogeneity caused by the ever-increasing volume of data provided by the internet. Personalized recommendations can be made using computer programming based on a user's past preferences, just like a search engine that looks for and identifies objects in line with the keywords or symbols nominative by a user. An advice system's goal is to offer customers with relevant information about a product in a timely manner so that they may make informed decisions. To accomplish this, the system will automatically filter out irrelevant products and recommend only those that are truly relevant. Triple-crown applications of recommender systems can be found. Product recommendation in E-commerce websites, motion picture and video recommendation, news recommendation, App recommendation, music recommendation and social recommendation are just some of the many best-known examples of recommendation.

Generalizations: We can find mainly three types of recommendation system techniques as Collaborative Filtering [1] [2] , Content Based recommendations [3] [4], and finally Hybrid recommendation. Among them most popular concept is collaborative filtering. It assists individuals in making decisions based on the ideas of others those who have same desires to their own. It is possible to recommend things that are identical to those that have been previously liked by a given user using the content-based recommendation technique (CB). Each recommendation technique has benefits and drawbacks of its own. As a result, hybrid recommendation strategies are presented in order to achieve improved performance while avoiding the shortcomings of pure given request. [2] To overcome problems concerning cold start, sparseness, and/or scalability, the most typical hybrid recommendation methodology is to mix cluster analysis with other recommendation techniques.

1.2: Introduction to recommender systems

Researchers have concentrated on the Recommender System (RS) as a guide to assist people find items directly by continuously providing recommendations. There is a wide range of information retrieval strategies that are addressed in this review of the extant literature. The primary goal of recommender systems is to assist customers in making an online purchasing intention by offering excellent recommendations [5]. Over the last few years, academics have focused on enhancing the application and performance of RSs [6]. For my literature review, I have paid my attention on many of recent world popular research paper sources. This would comprise peer-reviewed research articles and proceedings from relevant conferences. There are a variety of ways to the recommendation that have been implemented in various contexts. The most common recommendation/filtering methods in RSs are described in this section. Each of these methods has its own input data source, extraction procedure and constraints, as outlined here in terms of the most prevalent recommended approaches like CFRS, CBRS, DRS, KBRS, and HRS. may be found in the following Table:2.1

1.3: CDRS (Cross Domain recommender System)

CDRS models will execute with multiple domains. Those are the D_S and D_T. And D_T can extract more knowledge from D_S. It may lead to providing more accurate recommender results. As well as, we can say in CDRS, the D_T can learn from the source domain's (D_S)data. When we come to the CDRS building structure, there are main three stages. The first domain stage. That means we need to identify or create the target domain and the source domain. After that two domains are available now, we can perform our next recommender task. So, what is the next stage in CDRS? The second stage is we call it item and user overlap scenario.

Once there is enough user and item overlapping then I can perform our recommender tasks well. The last stage is the recommender task stage. As we all know now, if we have all these major three stages then we can execute or run our CDRS model and we can evaluate it later. In the recent world, we can see dingle domain recommender systems are struggling with many issues since user expectations and desires are rapidly increasing with high variety. Some of them are new user-item problems, data sparsity issues, low diversity, novelty, and less accuracy. But CDRS can use to find solutions for those kinds of issues. Cross Domain Recommender System is one of the not much-explored areas and it needs more research works and there existing many research gaps. Cross Domain Recommender system model's most general purpose is for measure accuracy using mastics like RMSE, MAE, etc. Since the more need than accuracy to maintain high-quality recommendations, we need to pay attention to a few specific areas beyond accuracy like diversity and novelty.

CDRS has gained traction within the last few years, and academics from a variety of backgrounds have begun contributing. Some of the strategies that previous researchers performed are mentioned below.

- In order to determine the link between source domain and target domain researchers have used graph-focused approaches.
- Also, we can use tag-based methods to create metadata association between both domains.
- Semantic-based approach for the development of knowledge

Accordingly, we can conclude that these main three key characteristics are the fundamental units of CDRS, as presented by several researchers in papers [7].

There are four basic aspects to the understanding of the CDRS scenario. The first one is domains. As an example, we can see the target domain and source domain. As per our use case we have to identify carefully what is our source domain and what is our target domain like that. The second one is our business goal. To achieve those respective goals, we can apply proposed cross-domain models. The third part is the recommender task. It senses we should consider what are the data set that we should use our CDRS model. The last part is called scenarios. Here what we should consider is we need to identify which type of overlap method that we can easily apply between our identified domains.

In a nutshell, let me go through the most important things.

An area of thinking, activity, or interest is known as a domain. Researchers have looked at a variety of domain concepts like movies and songs, books, and movies like that. We know that domains are distinct from one another since they contain a wide range of products, such as books and songs. Alternatively, it has a variety of user kinds, such as those who pay yearly and those who pay per view.

Anyway, here is what we are going to focus on only two domains

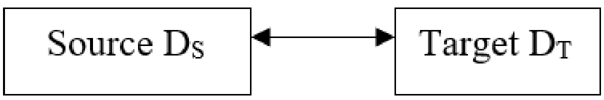

Figure 1. 1: CDRS - focused domains

We are able to find that there are multiple domain levels in CDRS:
As an example, let’s see

- under the item level, section we can find music Vs movies:
  Ex: Last.FM and MovieLens
- under the attribute level, we can see Thriller Vs Comedy
- Also, under the type of level, we can find Books Vs Movie

1.4: Research gaps and problem

The recent CDRS trending concept is introduced by researchers to the world to achieve many interesting goals in the environment recommender systems. CDRS is one of latest section in main domain called RS and still there are many research gaps and challengers and lot here to be developed and need to perform more future research works in CDRS domain.
One key issue that I found in the CDRS open research issues pertains to the metrics used to evaluate the suggestions. General accuracy metrics like as MSE, RMSE and MAE that measure the difference between anticipated and actual ratings, are commonly used to assess the usefulness of CDRS. However, the problem will be the accuracy value isn't high enough to generate CDRS recommendations that are both useful and of high quality.
Only a few of the finest recommendations are displayed in commercial systems, whereas forecasted ratings are not. Thus, the system recommends some things that it believes will be popular with its customers. Based on classification metrics or ranking metrics, direct evaluation of the performance of the top N recommendations must be achieved.
If we want CDRS recommendations that are meaningful, we must focus on metrics like diversity and sparsity as well as the more widely utilized accuracy in order to get the best results for our consumers. Cross-domain recommendations, as one might anticipate, are less reliable than focused on the equal level of data relevant for D_T. While CDRS may be more reliable, the actual advantage is in the diversity and novelty they provide, which can help to greater user trust and utility.

Diversity metric as a focus throughout this study, I plan on creating better recommendations in CDRS that are also highly rated by its users. Researchers in the recommender system field have long been concerned about overfitting, but diversification has also evolved as a technique to enhance the user's experience. Diversification provides a lot of space for innovation because it is such a fresh concept. Overcrowding and a lack of security can be combated by a diverse population. RS-related difficulties require significantly more human involvement than the over-fitting problem, making this a fascinating read.

1.4.1: Research Gap

- Currently, most Commercial Recommender systems are suffering from low recommender quality with increasing user dissatisfaction. So, it's clear only the prediction accuracy is not sufficient to measure the recommender quality of the Cross-Domain Recommender Systems.

1.4.2: Diversity

- It's an exciting time to study diversification because it's still relatively new, thus more research and new improvements in RS, especially in CDRS, are necessary.
- Diversity is primarily determined by user expectations, and diversity enables RS to meet the needs of a greater number of users' needs and expectations. Diversity aids in considering user behaviors and changes in their RS interests. Different similarity metrics generate different diversity metrics, which vendors greatly value.

1.4.3: Why is it necessary to conduct RS research with a diversity focus?

- Only a few studies, on the other hand, have systematically analyzed and compared the diversity performance results in recommender systems. As they explore other users' profiles for prediction, CF recommenders can lead to more diverse item lists. So here is how I attempted to conduct my research on diversity-based CDRS using the CF method.

1.4.4: Why is diversity also so important along with accuracy?

- When recommending items to users, it is pretty much worth considering many performance metrics and not just the accuracy of a rating.
  Ex:
  Allow us to put ourselves in the shoes of someone utilizing a trip recommendation system. So, let's pretend that all of the suggestions it makes are for locations we've already been. It doesn't matter how accurate the RS system is at rating and predicting, because we've ranked all the places we've been in the order of choice. As a result, a poor recommender system will be the result of this behavior. It senses that the user expects for some randomness or diversity in the outcomes of the recommender.

  So next time would you like to use such RS? most probably our answer is No
  That's why we should pay our attention on more than accuracy in future RS works.

1.4.5: Why diversity is important for recommender quality?

- To provide useful better recommendations for users via CDRS we must carefully pay attention to other specific evaluation metrics like Diversity and Novelty rather than commonly used accuracy.

1.5: Objectives

- The main objective is to implement a Cross Domain Recommender System (CDRS) via CF to enhance recommender accuracy by focusing on diversity.

- Moreover, I have constructed two SVD matrix factorization-based Collaborative filtering models based on explicit feedback on the MovieLens 100k data set in this study. The general collaborative filtering model and the cross-domain model are among those couple of models.

2: LITERATURE REVIEW

This section focuses on studies that are related to this research. All the up technical, such as recommender systems and CDRS, are covered in this introduction. Under title 2.1 discusses the major prevalent forms of recommendation systems, while Section 2.2 details the most common issues that arise when implementing one of these systems. Section 2.3 also discusses information retrieval methods used in rec-systems. CDRS and their difficulties and primary assessment tactics are described in this section 2.4.

2.1: Recommender systems

The rapid expansion of the World Wide Web is leading to increasing complexity of knowledge present online. Each user has unique desires and expectations, and they can experience lot in web by using varies kind of applications. Users can use various kind of recommender applications to enhance their personalize user expectations and they can get new knowledge from RS, to improve their own desires. RS are a special kind of information filtering systems. As a result, they provide recommendations to the user based on their stated or implied preferences. There are many types (Table 2.1) of recommendation systems are using today in enterprise level business industry specially in e-commerce, e-government, and e-learning.

2.1.1: Collaborative filtering recommender system (CFRS)

In the Recommender systems domain, the most prevalent filtering method has been the collaborative filtering approach. This method has been adopted by a large number of researchers. In addition, a lot of effort has been put on minimizing the difficulties of employing CF in RS. With an emphasis on CF's issues [8], the research article illustrates how various forms of filtering are utilized in recommendation systems. CF, CBR, and Hybrid recommendations are three of the most common methods for making recommendations to clients, are also addressed in another study [9].

There are two categories come under CF:

- Memory Based collaborative recommender system (CRS)
    - Memory Based CRS's two primary phases are prediction computation and similarity measurement, and they are further divided into two components based on how they compute similarity
    - Item Based CRS: A set of items is used to do the similarity computation [10]
    - CRS based on user similarity: The similarity values of users are used to compute the similarity [11].

- Model-based collaborative recommender system (CRS):

    - Model-based CRS [12], It uses a variety of ml algorithms, including Bayesian networks, clustering approaches.
    - Despite its effectiveness, model-based CF still suffers from data sparsity. Users and items with lower ratings will receive inaccurate suggestions.

2.1.2: Content-based Recommender System (CBRS)

CBRS to make suggestions focused on a user profile and the description of the item. For example, a user's previous search or purchase history may be included in their user profile. Any suggestions based on past user preferences will be taken into consideration by the system because of this. The similarity between the two items is then determined by comparing their respective features. Mainly focused on user's positive feedback for comedy, the system will be able to recommend other comedies. Users' interests can be better matched by content-based recommender systems because semantic material is used in their construction of the recommendations. It is possible to recommend new and unpopular things using content-based recommender systems.

Additionally, it can include a list of content-features for which an item should be recommended, hence assisting in the categorization of recommended products. Collaboration filtering is not concerned with sparsity because it does not require knowledge of other users' preferences when making recommendations. Overspecialization is a significant concern for recommender systems that are content based. The only criterion that may be used to produce suggestions is the user's preferences; it cannot recommend products that are not included in the user's profile. [13]

2.1.3 Knowledge-Based Recommender System (KBRS)

It's possible that not enough information is available or relevant (even if it is) to make recommendations for less frequently purchased items like Television sets, bicycles, and the like [6]. Extra details are designed to fulfill this task Knowledge-based RSs make recommendations based on extra information about the user-item relationship. [14]

A type of constraint-based recommendation system (RBS) uses a user's preferences to determine what products to recommend based on the user's preferences [15]. If no such item is available, a list of possible alternatives is provided. Data from a user's profile and item description is matched up with the recommendation to ensure a good fit.

2.1.4 Hybrid Recommender System (HRS)

Combining multiple filtering techniques results in Hybrid RS. CBS and CFRS are the most popular Hybrid RS pairings. The goal of using multiple CF methods together is to enhance recommendation accuracy while minimizing the limitations of specific filtering strategies [16].

Table 2. 1: Summary of Recommendation Types

Rec-Approach	Source of data used in the approach	Extraction model	Limitations
CF	User feedback on products (explicit or implicit)	Cosine and correlation based similarity and K -nearest neighbor,	user's rating problem, Sparsity problem, Synonym problem and false and Cold start problem
Content-based	Explicit or implicit data provide by users.	Similarity metrics, Information filtering and document modeling	Non-homogeneous items and no methods for predicting missing ratings, Content analysis issues
Demographic (DRS)	Demographic information about the user, which including gender, age, date of birth, and so on.	locating group interests	It is entirely dependent on demographic data, which can be inaccurate and static in nature, among other things.

Knowledge based (KBRS)	Information on users obtained from social media sites, data gathered during a search or request for a product, and so on	Mainly ML based models and Decision rule.	Knowledge gain requires a higher cost, and preferences are not necessarily independent of one another.
Hybrid (HRS)	Any of the sources that are listed in this table are acceptable	Any kind of extraction methods that are listed in this table are acceptable.	Recommendation novelty, coverage, and serendipity

Table 2.2 provides a list of the most common filtering methods used during the most common application domains of RS.

Table 2. 2: Filtering Techniques with Application Domains

Application	Filtering approaches used
e-library and e-learning	Content-based, Collaborative, Hybrid Knowledge-based
e-government	Knowledge-based, Collaborative, Hybrid
e-commerce	Knowledge-based, Demographic, Content-based Collaborative, Hybrid

It has also been the most widely used filtering method in RS, using collaborative filtering (CF). This method has been used by a large number of researchers. In addition, significant efforts have been made to minimize the difficulties of utilizing CF in RS. Other common use cases for recommender systems include e-commerce, government, and education.

2.2 Recommendation problem

Some of the most pressing issues with recommender systems are shortly addressed in this section. Table 2.3 has summarized all those big problems below. [14]

Table 2. 3: Approaches to making Recommendations that have Significant Flaws

Filtering approach	**Analyses of only a limited scope**	**Over specialization**	**Cold start**	**Sparsity**	**Scalability**	**Long tail**
Collaborative (CFRS)			✓	✓	✓	✓
Content based (CBRS)	✓	✓				
Demographic (DRS)	✓	✓				
Knowledge based (KBRS)					✓	
Hybrid (HRS)					✓	✓

2.2.1: Over-specialization

- Admittedly, some algorithms may certainly do the opposite of what they've been supposed to. Since CBRS doesn't individual considers from a derived from a non-homogeneous group of things, this results in lower accuracy. To fix this problem, new hybrid strategies that enhance the accuracy of the recommendation process must be constructed. Search results are whittled it down by utilizing content-based filtering learning methods that appear for the most relevant documents in the database. The user, however, is reduced to documents that are comparable to those previously seen [14].

2.2.2: Cold start

- No past records (ratings, interests, search history) are readily accessible for an RS to use when recommending an item or user to the system [15]. Cold start problems are what they're named. In addition, it is considered as an emerging user issue or the new item issue.

2.2.3: Sparsity

- In the real world, the RSs deal with very large datasets on a daily basis. That's why collaborative filtering uses an extremely sparse user-item matrix, resulting in an unsatisfactory prediction or recommendation results from CF system models.

- It can also occur when a person has used a product but has not taken the time to rate it after using it. Occasionally, users will not rank goods that are unfamiliar to them. This difficulty is addressed by RS through the use of a methodology known as the clustering method. [17]

2.2.4: Scalability

- In the event that a large number of things are left unrated or have a poor rating, this would be happened. The recommendation library will lose an item over time if it is first low rated or not evaluated at all in an RS after a top-N suggestion. This issue is intimately linked to diversity. It underlines the need of providing users with a wide range of options and the wide range of differences between those options. This element, however, is ignored by RSs, resulting in the long tail issue [18]. Figure 2.1: indicates the visualization of recommendation problem.

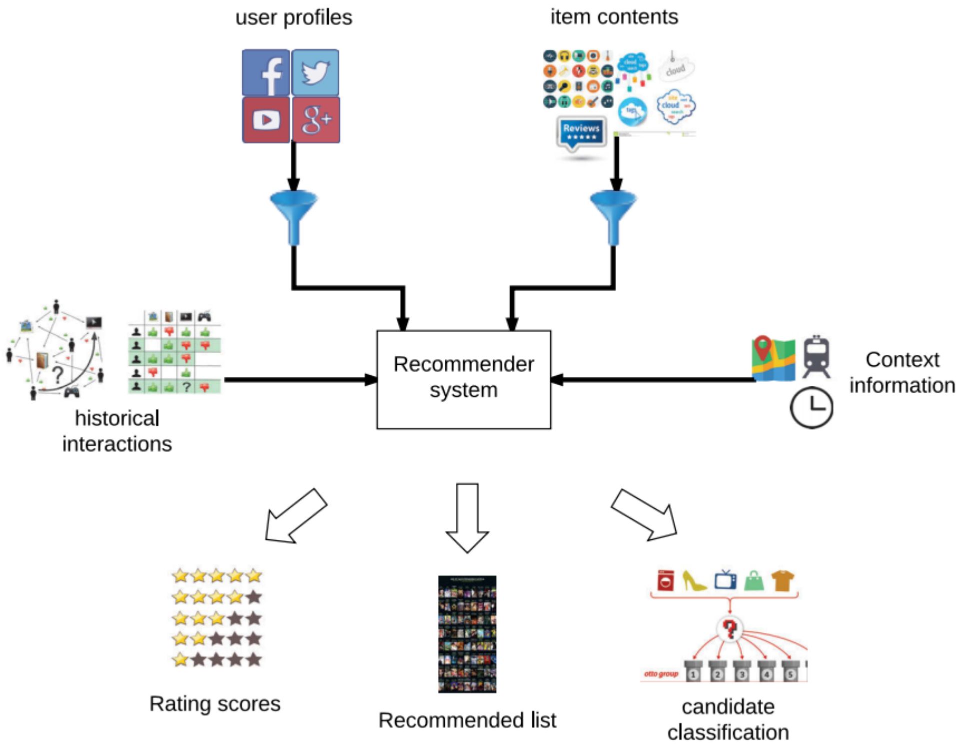

Figure 2. 1: Recommendation Problem

2.3 Information Retrieval Techniques in Recommender Systems

- **Logistic regression:**
  It is possible to predict discrete variables using both continuous and discrete data using logistic regression. For example, E. Montanes et al. have used a collaborative tag RS to rank the most relevant tags in social networks.

- **Machine learning:**
  Machine learning is the process through which a machine may learn by itself, without requiring to be explicitly programmed. For instance, it utilizes a variety of approaches like logistic regression and decision trees, as well as clustering and Bayesian networks.
- **Decision tree:**
  Using a decision tree is an important strategy for restricting the field of possibilities. It is utilized in RS to evaluate and forecast the absence of references from users. As a result of Qi Yu's work on the cold start issue, new items now have an outstanding service recommendation.

- **Association rule learning:**
  Users and items data are analyzed using association rule learning to identify the most common patterns and associations that can be applied to generate a recommendation. An association rule mining approach combined with both CBF, and CF was being used by Tewari et al. in order to calculate how prospective buyers will be in their book recommendations. [18]
- **Bayesian network:**
  A probabilistic model known as a Bayesian network classifier is used to classify large networks like social networks. To overcome cold start issue, [19] proposes a trust-based probabilistic recommendation strategy.
- **Matrix or Tensor Factorization:**
  An example of a latent variable model is a matric factorization. In the Netflix Prize Challenge, this algorithm got widely known. Both people and things are categorized into factor vectors to represent the rating patterns at the foundation of this technique. Additional data could be fed into the algorithm. It is in this wiggle room that new information can be found and explored. By employing uninterested items, Lee et al. [20] came up with an RS where users' interestedness values were calculated and inserted into the sparse rating matrix.
- **Deep learning:**
  Deep learning can be applied as new trending research area by enabling to discover hidden patterns in data. Effective and dynamic behavior models can be built using it [21]. We can learn a lot about a user's attitude by examining the neural network's supervised and unsupervised learning methodologies. Recommendations on YouTube have been delivered by Paul Covington et al [22] that used a deep neural network. The user's input is placed into a deep learning algorithm is performed to rate the movies that are most relevant to the user. It has been observed that using a deep learning approach, Ali Elkahky et al. [12] can help maximize the relationships between users and their favorite objects in latent space.

2.4: Cross-Domain Recommender System

This section aims to aggregate current research by first identifying CDRS characteristics, emphasizing numerous definitions connected with many features, and detailing how researchers have defined Cross Domain recommender System.

2.4.1: Definition of Cross-Domain Recommender System

In general case CDRS consists with main two domains called Source domain and Target domain and those run a vital role in cross-domain recommendations in enterprise level business use cases. As per the past research papers specially from paper [23] and [24] they have explained how knowledge transferring or knowledge extraction happen between those target domain and source domain. It was common knowledge among moviegoers of many subgenres. We're able to convert binary knowledge into numerical ratings of the same people, products, and things that they engage with. According to [25], matrices with time stamp differences across consumers and the same items can communicate information. Primary studies can be classed based on the advice they offer. Both [23] and [26] generated recommendations for users in both the target domain, which was MovieLens [27], as well as those in other domains. That's why this type of task is called "recommendation" because both the source and target users might be involved. CDRS models according to literature, have a lot to offer in terms of domain difference, overlap between users and products, and recommendation duties.

According to Ivan Cantador [7], there are several aspects to be considered about domains. Their findings are mentioned in below.

- Domain definitions from paper [7]

    - Item level: Most of the attributes of things are shared by domains at the item level. Films and books, for instance, may have identical titles, but they are two quite different mediums.

    - (Item) Attribute level: If two items' attribute values disagree, they are regarded to be in diverse disciplines.

    - (Item) Type level: If some attributes of two items differ while others remain the same, they are said to be in distinct parts. While television and films may have a title and a genre, other characteristics, such as run time, set them apart as separate genres.
- Li's domain definitions
    - Data Domain: User interactions with things may be recorded as binary ratings, like "like" or "dislike" ratings based on the item in question. This is a common practice in RS. The acquisition of multidimensional information for similar encounters involves several data domains. [28]
    - Time Domain: A Time Domain can be constructed by dividing a rating matrix with time stamps into discrete intervals. Each time slice is then considered as a separate temporary domain. [29]
    - System Domain: Every RS is treated as a domain in sparse recommender system rating matrices. Learning transfer from a high-rated domain to a low-rated target domain. [30]

Overall, there are two sorts of cross-domain techniques (figure:2.2 and figure:2.3), depending on how the source domain's information is utilized.

01: Linking/Aggregating knowledge

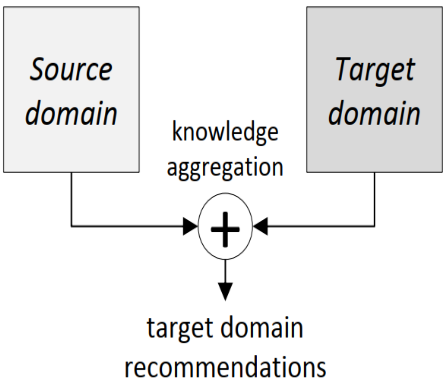

Figure 2. 3: Linking knowledge

02: Sharing/Transferring knowledge

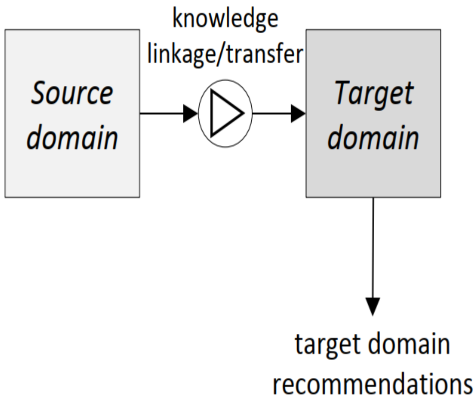

Figure 2. 2: Sharing knowledge

- Linking/Aggregating knowledge
  - Bringing together user interests
    - Aggregate user preferences
    - Ex: Tags, click data, transaction logs, ratings
    - Well, performed for new user issue

  - Providing a way for users to model data
    - Combined models from a variety of disciplines as CBF, CF and Hybrid
  - Combining recommendations
    - Aggregate single-domain recommendations
  - Connecting domains
    - Connecting domains by a commo utilizing a shared understanding to link diverse fields

- Knowledge exchange and transfer
    - Sharing latent characteristics
        - Shared latent attributes link the target and source domains.
    - Rating pattern transformation
        - Rating styles are transferred between domains.

According to the conference on 8th ACM Conference on Recommender Systems in 2014, Paolo Cremonesi and Cantador has nicely explained the cross-domain recommendation tasks in figure 2.4.

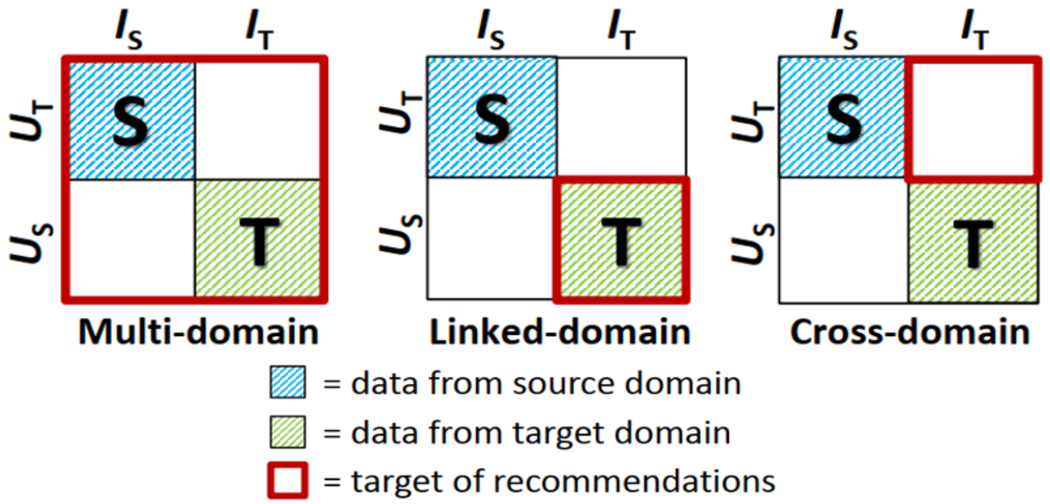

Figure 2. 4: Cross-domain recommendation tasks

Furthermore, several specifications of each mentioned cross-domain recommendation tasks are illustrated below concisely.

- Multi-domain recommendation task
    - Recommend items in both D_S and D_T.
    - Goal: cross-selling, serendipity
    - Approach: linking domains and sharing knowledge.
- Linked-domain recommendation task.
    - Recommend target items to users in the target domains.
    - Goal: improve accuracy of recommendations in the target domain (e.g., reduce sparsity)
    - Approach: all
      (Linking, Aggregating, Sharing and Transferring knowledge)

- Cross-domain recommendation task
    - Recommend items in the D_T to users in the D_S.
    - Goal: solve resolve the cold-start problem, new users and new item problem.
    - Approach: Aggregating knowledge

2.4.2: Concerns Associated with cross-domain recommender system.

Some common issues in CDRS building blocks were discovered by referring to previous literature works. However, consumers can only score a small number of things in real-world recommender systems, therefore the rating matrix is always sparse. For probabilistic modeling, k-NN search or matrix factorization the currently offered rating data is drastically insufficient. For most CF approaches, the sparsity issue has become a great obstacle. Although we are unable to fabricate additional ratings in the considered rating matrix, we can borrow appropriate information from a rating matrix in a different domain.

Consider the following: Book rating website opens. Due to a dearth of visits at the start, only a few ratings can be used. Let's imagine a well-known movie review website already has a comprehensive matrix of movie reviews. So, can we build a bridge between the two rating matrices and transfer useful rating patterns from one to the other? Because movies and books are connected (in genre and users of both rating services may reflect comparable societal elements) [24]

Most CF approaches suffer from the sparsity issue. [24] In concept CF, user item rating pattern deviations from other domains to a target domain sparse rating matrix. They didn't even demand that the two domains' users and stuff be similar.

In view of the target matrix's limited ratings, this bridge has been constructed between the two rating matrices at the group level of customer rating patterns. To begin, they've condensed the ratings from the auxiliary rating matrix into a codebook, a representation of rating patterns at the cluster level that is both instructive and condensed. As a result, they have presented an efficient approach for reconstructing the goal rating matrix by increasing the codebook. As well as, they had carried out a large number of empirical tests to demonstrate that their approach is superior in terms of handling the data sparsity issue through the transfer of useful knowledge from the auxiliary tasks to the primary work.

2.4.3: Factorization Machines for cross-domain recommender system

The use of Factorization Machines (FM) as a powerful and adaptable model structure for collaborative filtering scenarios is a good place to start. Matrix factorization has been used successfully to implement a very small number of latent factor models. Matrix factorization, in its most primary form, describes alike things and users using vectors of factors defined from item rating styles. If there is a high level of agreement between the user and the attributes of the item, a recommendation may be made. These approaches have grown in popularity in recent years due to their ability to be easily scaled. [31]

Mathematical models called Matrix Factorization (MF) [32] attempt to understand low-dimensional interpretations of item and users in common latent variable space. As a result, we can approximate the product of two low-rank matrices that include item and user embeddings in our sparse, observable user-item interaction matrix. Latent factors will be used to calculate the similarity between items and users, as well as to infer preferences that cannot be observed.

It is through the minimization of the squares of the errors between the observed and anticipated ratings that MF algorithms learn about any of these people or items' hidden variables. In order to avoid overfitting, several specifications also incorporate global item and user biases and regularized terms. FM models capture user-item interactions as tuples of real-valued feature vectors and numerical target variables and should be familiar to anyone who has trained a standard regression or classification model.

As the foundation for collaborative filtering, binary vectors of user and item indicators are commonly used, so that each training sample contains precisely two non-zero entries matching the given item and user pair. These indications could be supplemented in a variety of additional elements, such as item or user properties and contextual information about the engagement itself. Figure 6: depicts the construction of a simple MF model.

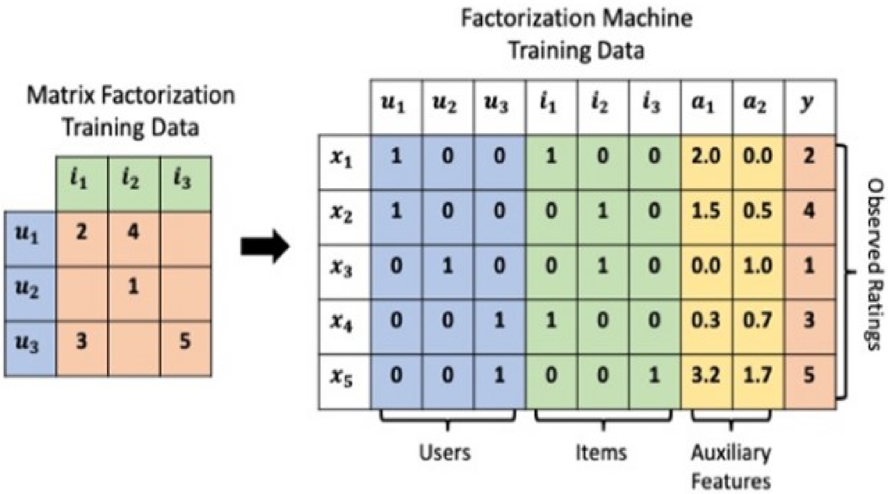

Figure 2. 5: Factorization Machine

3: METHODOLOGY

In this research study, I have built a couple of matrix factorization-based models via collaborative filtering method using explicit feedback. One is general collaborative recommender system and meanwhile other one is cross domain recommender system; both are based on well-known recommender system research data set called MovieLens 100k data set. So here in this section firstly I have discussed about the data set used here and later I have explained how I build these two models and what are the technologies and methods used here to achieve the final expectations and goals.

3.1: MovieLens 100K Data Set

The MovieLens 100K dataset was used in this research. It is a collection of data from the MovieLens website that has been compiled over time. Inconsistent benchmarking data.

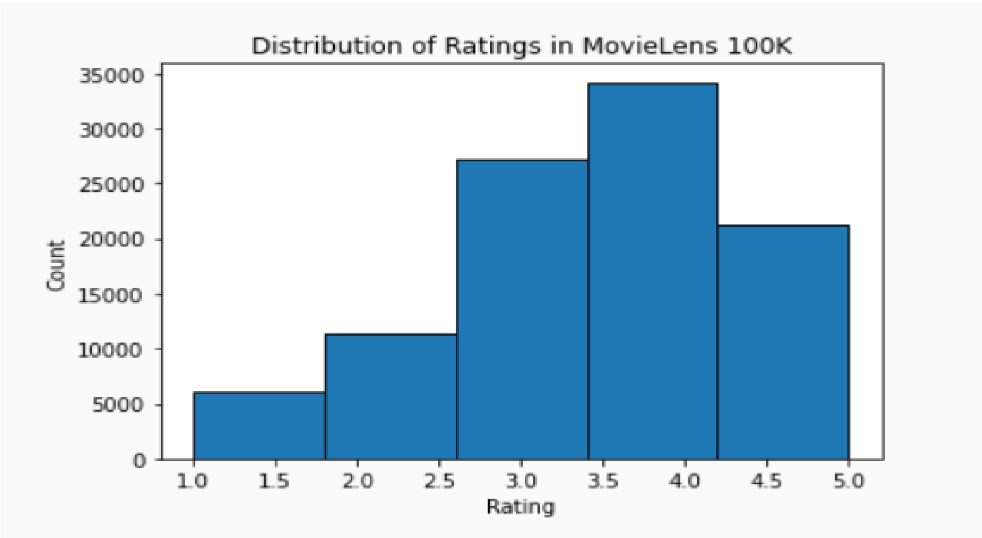

Figure 3. 1: Distribution of Ratings MovieLens

- Furthermore, we can find a few more components in the MovieLens research data package.
- u.data is for entire u data set, with 100000 ratings from 943 users on 1682 items. At least 20 movies have been rated by each user.
- All the users and movies have been ordered in sequential order starting from 1 and there are arranged in randomly
- User id, rating, item id and timestamp are separated by tabs.
- We are able to find user ratings details from u.info file.
- We are able to get information of the movies from u. item.

- There is also a tab-separated list of popular movie genres such as comedy, crime, animation, documentary, actions, drama, war, fantasy, horror, musical, mystery, adventure, children's, romance, sci-fi, thriller and western.

3.2 Applying Surprise python library (Initial steps)

Generally, when we are working on data science projects, we all use default libraries named Pandas, Scikit-learn, TensorFlow, Pytorch and so. Later Surprise came into the battle as a new open-source Python library most specifically developed for recommender system implementation tasks. We can use this library to build our own recommender models using user explicit data like user ratings. In the Surprise architecture, there is a python base class named “AlgoBase”. Here I have performed object-oriented methods to build my models.

So that base class has a set of functions and variables. So, we can easily share those between other classes which inherit from that base class. First, I created a new class that derives from the base class. Suprise library structure has been developed to predict our explicit data for every film or movie for every user and later after the successful model execution, it will provide us the prediction results, for example, it will give top-N recommendations to us. Mainly here we are performing the offline evaluation. Moreover, next, we can create an estimate function, which takes three arguments. It is a reference to the instance of the object.

After that, we can access all the variables associated with that instance. We may quickly and easily access any variables associated with an instance of this object by using the instance reference. When the estimate function is called by the surprise library, it will predict rating values for its users. It should be noted that the IDs for these users and items are known as "internal IDs." The raw user and item IDs in your source data must be mapped back to them for them to work properly.

In my research study, I planned to use some different kinds of evaluation protocols or simply called evaluation matrices. For that case scenario, I implemented another class (RecMetric), that we can use all the evaluation metrics on it. Simple this class made it easy for me to calculate novelty, diversity, and accuracy. When I call evaluate the function, it evaluates each eval algorithm, and prints out the results. The beauty of this is that I don't even have to use the EvaluatedAlgorithm or EvaluatedData classes when we want to start playing around with new algorithms and testing them against each other. All we should need to do is use this Evaluator class, which has a simple interface. We are making all the trouble of developing this CF-based general recommender framework so we can do this.

Since in recent times Singular Value Decomposition (SVD) is one of the best algorithms available right now for matrix factorization, SVD has been used in both the CF model & CDRS model to overcome the data sparsity issue and achieve better results.

3.3 Neighborhood-based collaborative filtering

[Both models have been implemented via this technique]

This is the concept of using other people's behavior to inform what we might enjoy. At a high level, it means finding other people who are similar to you and recommending things they liked, or it could mean finding other things similar to something we like. That is, suggesting items purchased by people who also purchased the items you liked. The idea is to take cues from people like us, our neighborhood if you will, and recommend things based on what they like that we haven't seen yet. That is why it is referred to as collaborative filtering. It makes recommendations based on the collaborative behavior of others.

Steps for Collaborative Filtering:

- We start with a data store of some sort that includes rating information, mostly like explicit feedback from our users.
- Then, by looking up other items like the items each user liked, we can generate candidates for things to recommend.
- Next, we score and rank those candidates, filter out the stuff the user hasn't already seen, and produce our final top-N recommendation list.
- There are several methods for determining the degree of similarity between people or things. The possible to identify user like us or items similar to items we have liked is at the core of neighborhood-based collaborative filtering.

The first step in collaborative filtering is measuring the similarity between things or the similarity between people. That sense we can find people like us. There are lots of ways to do this. The cosine similarity metric and Pearson_baseline matric work well in most cases.

3.4 Train/test splits & k-fold cross-validation

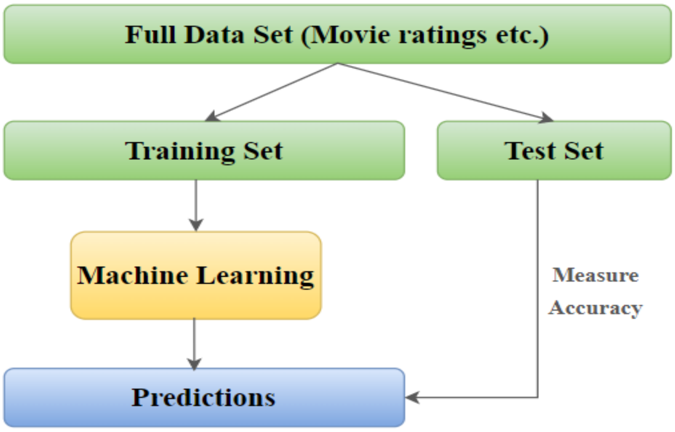

Figure 3. 2: Train and Test Splitting

- A recommender system, as we all know, is a machine learning system. We should train our recommender model based on previous user behavior or simply sense previous user ratings.
- Then we use it to generate predictions about items that new users might be interested in.
- We can evaluate the performance of our recommender system to predict how people rated things in the past. To begin, we divide our ratings data into two sets: training and testing. In general, the training set is larger, containing 80% of all our data and 20% of the data for the test set. (Figure:8)
- We only use training data to train our recommender system. That's where it all learns the needful relationships between users or items.
- The above procedure, which I have performed when I was building a general CF-recommender model. Once the training stage is done then we can perform a prediction stage about how a new user might rate some item they have never seen before.
- For that case scenario, I used the data that we reserved for testing. To improve single train/test split I have tried k-fold cross-validation in my general CF-Model.
- It is similar to train/test, except that rather than just a single training set, I created numerous randomly assigned training sets.
- Each training set, or fold, is used independently to train our recommender system, and then I can evaluate the resulting systems' accuracy to our test set.
- And finally, we could get the average accuracy score among all the K-folds.

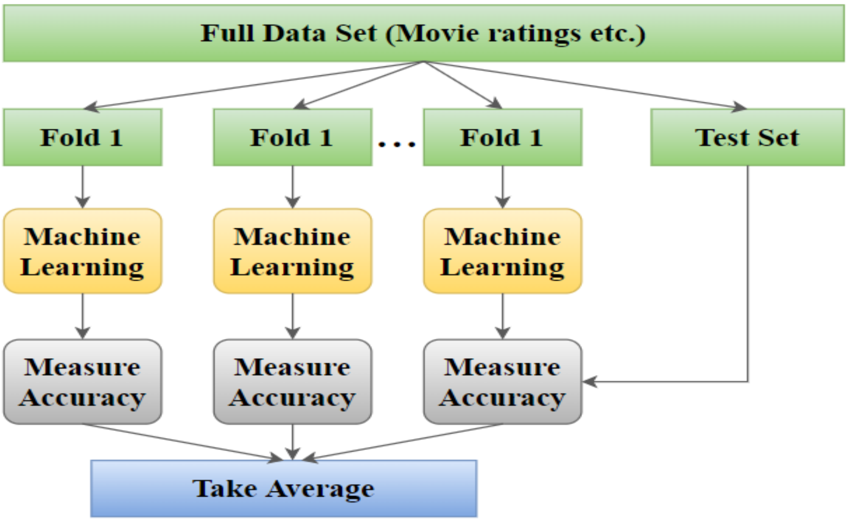

Figure 3. 3: K-Fold Cross-Validation

3.5 KNN Baseline

- KNN baseline is yet another KNN algorithm that takes into account user preferences. KNN baseline uses users' baseline ratings rather than the mean difference rating or the z-score normalized rating (see the section on the Pearson-baseline correlation coefficient for more about baseline ratings).

- KNN baseline then compares the baseline ratings of the k nearest neighbors to determine the difference between their baseline rating and their rating of item i. These differences are then used to calculate the amount by which the predicted user's rating of I differs from the predicted user's baseline, as in KNN basic. Finally, the predicted rating is calculated by adding this amount to the user's baseline.

- In this research, my tests involved training a new sort of recommender known as KNN, and the best result was achieved by utilizing the KNNBaseline recommender in Surpriselib.

3.6 CDRS implementation

- As we know according to the definition of the cross-domain recommender system it can make recommendations using the target domain with the knowledge extraction from the source domain.
- So here first what I have to do is identify the target domain (which is we are going to train) and source domain (which is the target domain that can extract knowledge).
- Since I am using a single dataset here, so I tried to build the target domain (D_T)and source domain (D_S) via users. So, a set of users are included in the D_T and another set of users is included in the D_S.
- After that I did random splitting in the target domain and made target domain training set and target domain test set like that.
- After performing a few mathematical modifications on target domain training set
- And the test set has been modified. Then source domain is concatenated into a target domain training set.
- Major difference between the general Cf model and the proposed CDRS model is both have different type train & test set even both performed on the same data set.
- Simply general CF model has random train /test splitting, meanwhile, CDRS has Source and Target domain splitting mechanism.
- After designing each train and test in both models, then we can perform evaluation methods equally.
- Furthermore, here user-item overlap scenario is used to perform recommender tasks.
- The Following image shows the overall cross-domain recommender scenario

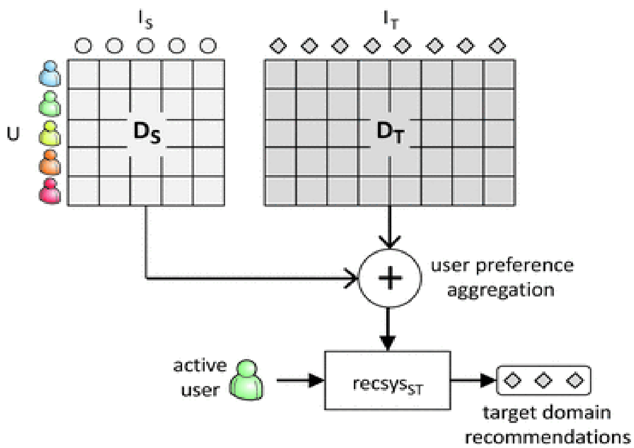

Figure 3. 4: Domain Identification-CDRS

3.7 Recommender model building workflow

Recommender models (Both General CF and CDRS) via Collaborative Filter (matrix factorization approach using SVD)

- Here I have used an open-source Python library called Surprise to build both models. First, I Must install that library.
- I got complete 100k rating data set from MovieLens and loaded up MovieLens data. Then I checked how is the popularity ranks since it is very useful to calculate novelty later.
- Next, I decided to build up the item-to-item similarity score then I can calculate diversity at the end. For that, I tried to train a different kind of recommender called KNN which can compute item similarity scores. However, for me, the final best performance came via the KNNBaseline recommender in Surpriselib.
- Then performed general train /test splitting in my general CF model and there randomly set aside 25% data for testing & 75% data for training purposes.
- This step is quite different within the CDRS model (cross-domain recommender system) because it has a specifically defined train set called Target Domain train set (Target domain train set + Source domain) and Target domain modified test set. In the CDRS model, we don't perform a random train/test splitting method which I used in the general CF model.
- The above step (Defining train set & test set) is the major difference between both models (general CF recommender model & Cross-domain recommender model) implementation in my research study.
- After defining training and testing data sets, the next task is building a recommender model. So, I have created a new SVD recommender algorithm with a fixed random seed so then we can get consistent results, and then trained the algorithm using the fit function.
- We then process our test set on this algorithm, which gives us back a set of rating predictions for all the test ratings that it was fed. Next, what should we do is measure how good those predictions are.
- To measure the recommender accuracy, I have used RMSE, MSE, MAE, and FCP functions in both the general CF model and CDRS model. That all evaluation methods are clearly described in the next section (Evaluation).
- Well, next to that I have to pay attention to top-N recommendations. For that user case, we need to apply to Leave One Out Cross-Validation. I have performed LeaveOneOut cross-validation, to withhold one known rating for each user, so we can see if we can successfully recommend that movie in our results or not. Then we must train our SVD algorithm again with left-out ratings for each user.

- So, once we have that set of predictions we can call the Top-N function and find a top-N list for each user. After that, we can measure the hit rate and point out the results. Next, we want to measure coverage, diversity, and novelty. These are all properties of the top-N recommendations and don't involve measuring accuracy.

4: EVALUATION

We hope to focus on evaluating the cross-domain recommender system and discuss its own results in this chapter. I ran the general CF and CDRS models offline with explicit user feedback. I have performed MSE, MAE, RMSE, FCP as major accuracy measurements. In addition, I have focused on hit ratio, precision @k, diversity, and novelty. And also, I have compared my proposed CDRS model's RMSE performance with a few of the recent popular recommender models, and the specifications of each model have been listed here. Let's see how the CDRS model's results and performance via this chapter.

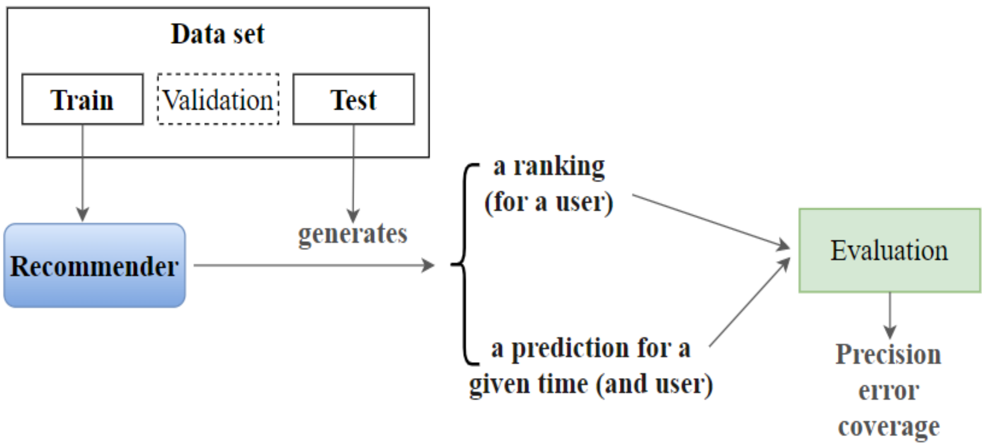

Figure 4. 2: Evaluation Process of the Rec-system

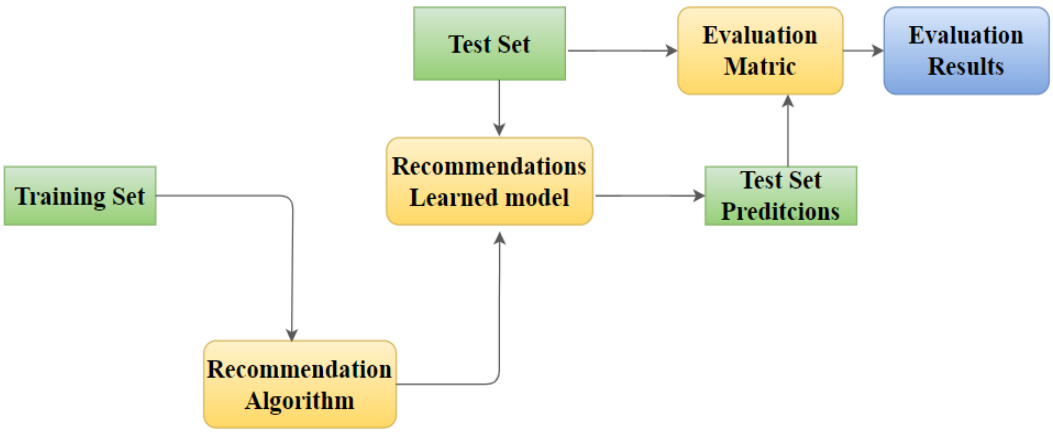

Figure 4. 1: Rec-System Overall Workflow

4.1 Accuracy metrics

4.1.1 Root-Mean-Squared Error (RMSE)

- So, its equation is like that of mean square error, the only difference is that we must get the square root. It is called the standard deviation of the error.
- This is always a positive value. And once it gives a low value that means it has high performance.
- It is beneficial to have lower RMSE scores because they measure mistakes.

Error = y - y hat (Actual value - Predicted value)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2}$$

Figure 4. 3: Root Mean Squared Error

4.1.2 Mean Squared Error (MSE)

- This is always a positive value; therefore, its result is closer to zero or a lower value is preferable to a higher value.

Predicted values are denoted by y hat
Actual values are denoted by y
Error = y - y hat (Actual value - Predicted value)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2$$

Figure 4. 4: Mean Squared Error

4.1.3 Mean Absolute Error (MAE)

- When the sample size increases, MAE will be a small value than the RMSE.
- Determine the average of all errors by taking the positive values of all the errors and summing them together.

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_1|}{n}$$

Figure 4. 5: Mean Absolute Error

4.1.4 Fraction of Concordant Pairs (FCP)

- Even this method is not much popular, I could find this from the paper called "Collaborative Filtering on Ordinal User Feedback" done by Koren & Sill [33]
- I have performed this FCP method to evaluate the recommender accuracy for both the general CF model and CDRS model.
- First, it uses pairs of ratings from the test set of the data.
- That sense every pair is two rated items from the same user.
- Then every possible combination of items that are rated by that user is a pair. Ex:
- If a user has rated items 1, 2, and 3, then we would have the pairs (1, 2), (2, 3), and (1, 3). Each pair is either concordant or discordant.
- For a pair (X, Y), if the user rated X higher than Y and the model predicted a rating for X higher than the predicted rating for Y, then the pair is concordant. Otherwise, it is discordant.
- Then we get the summation of the number of pairs from all users that are concordant and divide it by the total number of pairs.
- This is our FCP value, which makes sense since it is equivalent to the percentage of pairs that are concordant.
- Higher FCP value is indicated higher recommender accuracy.

Table 4. 1: Accuracy Results of CDRS and General-CF Model

Matric	General CF Model	CDRS Model
RMSE	0.8935932539	0.8783505664 ↓
MSE	0.7985089018	0.7714997175 ↓
MAE	0.6886527983	0.6738383664 ↓
FCP	0.6624670517	0.6636247302 ↑

4.1.5 Accuracy Comparison Results

Accuracy (RMSE) comparison with other popular industry research benchmark models which have been performed with the same dataset (Movielens100K).
The below table (Table:4.2) shows the results of RMSE performance in each model, how some recent popular research works, and my research works have been performed using the MovieLens100k dataset with using different techniques and tactics. Furthermore, in below chart (Table 4.2), I have added six columns like below.

"RMSE" Column = this column indicates the individual RMSE value of each model
"RMSE Improvement with relatively my approach" (Last column of the Table4.2) = In this column you can see, among the other benchmark models how RMSE has improved with comparing each other models with relatively my proposed one.
I have calculated the difference between each other model's RMSE value and my model's RMSE value. As an example, let's see the UDI-IS model in below table.
Difference of the RMSE = (UDI-IS RMSE value – CDRS RMSE value)
= (1.0135 – 0.878) = +0.1355
"+" notation indicates that there is a RMSE improvement in my model with comparing other model (UDI-IS) since we already know the smallest RMSE has the highest performance. Furthermore, I have made that calculation for other models also.

Table 4. 2: RMSE Improvement with Relative My Proposed Model

Model	User Feedback Type	ML / Deep Learning	Rec-Model Type	RMSE	RMSE Improvement with relatively my approach
UDI-IS	Explicit	ML	CF	1.0135	+ 0.1355
UCF-IS	Explicit	ML	CF	1.0082	+ 0.1302
ISCF	Explicit	ML	CF	1.0076	+ 0.1296
AutoSaved	Implicit	DL	Hybrid CF	0.901	+ 0.023
AutoSVD++	Implicit	DL	Hybrid CF	0.904	+ 0.026
mSDA-CF	Implicit	DL	Hybrid CF	0.8849	+ 0.0069
COFILS	Implicit	DL	CF	0.885	+ 0.007
Glocal-K	Implicit	DL	CF	0.891	+ 0.013
CDRS (My Model)	**Explicit**	**ML**	**CF**	**0.878**	**0**

As per the above table smallest RMSE value is belongs to my approach. Let's focus on more accuracy comparisons via following graphs.

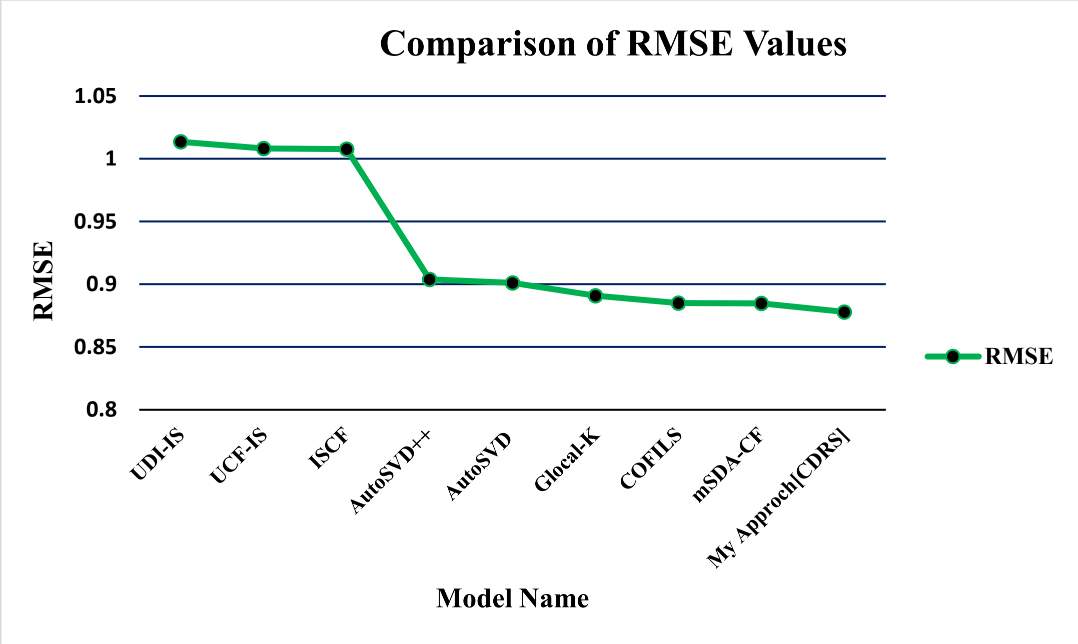

Figure 4. 6: Comparison of RMSE values with different studies

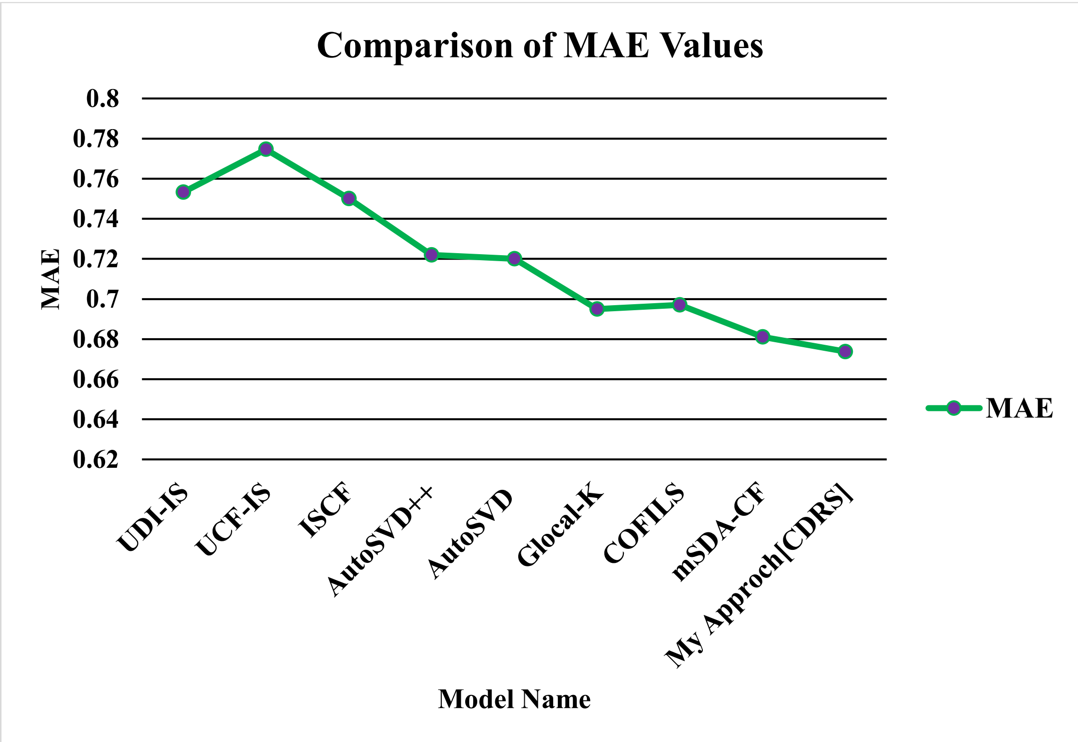

Figure 4. 7: Comparison of MAE values with different studies

Here also my approach has the best MAE performance.

Big Picture: RMSE, MSE & MAE results comparison all together

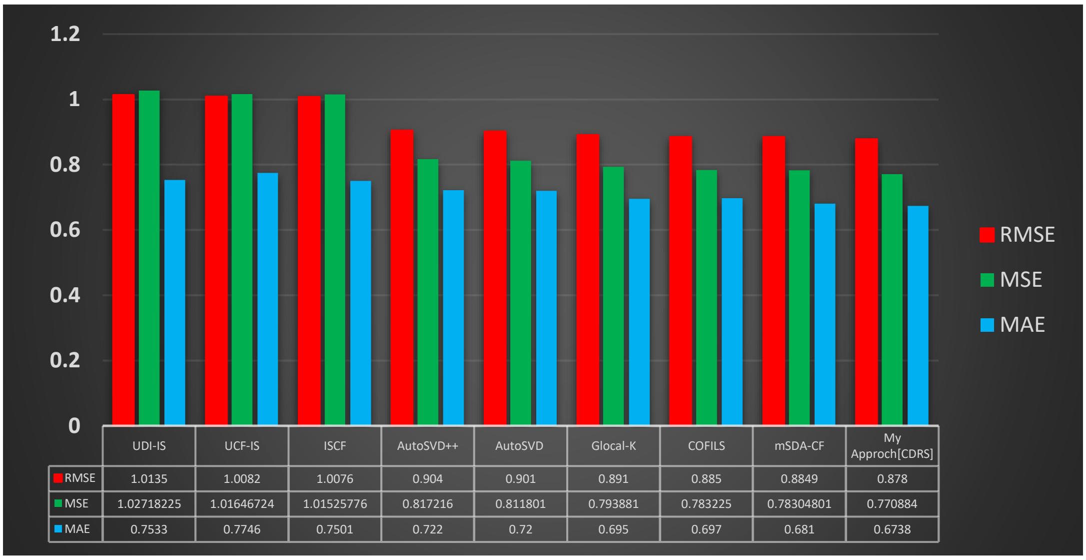

Figure 4. 8: Big Picture: RMSE, MSE & MAE results comparison all together

Findings from Figure 4.8:
Among all above models lowest RMSE, MSE &MAE values are achieved by my proposed approach.

Furthermore, I have added more necessary details for those benchmark models in the below table to get a clear idea of the variety of all those models with my proposed one.

Table 4. 3: Significant model's Specifications

Model	Specific used techniques	MF Technique	Goal
AutoSVD	contractive auto-encoder paradigm in conjunction with SVD	Conventional MF	Improving accuracy using implicit feedback
AutoSVD++	contractive auto-encoder paradigm in conjunction with SVD++	auto-encoders with probabilistic MF	Improving accuracy using implicit feedback
mSDA-CF	Deep architecture for CF by deep feature learning	Denoising stacked auto-encoders with probabilistic MF	Building, efficient ways to learn the effective latent

			factors via deep learning
ISCF	LCSIS -longest common sub is and ACSTIS-All common sub-IS	Sparse Covariance MF for Collaborative Filtering	To build new CF based method on user interest sequences
GHRS	Graph-based Features.	Probabilistic Matrix Factorization	Improving accuracy using the implicit feedback
Glocal-K	Performed two kernels called global and local.	A local low-rank sub-MF	Building both kernels for high-dimensional sparse matrix completion.
COFILS	Stacked Denoising autoencoder used	SVD++	Applying COFILS to reduce data scarcity
CDRS (My Model)	Surprise python library, KNNBaseline recommender, Pearson & Pearson-baseline similarity matric	Singular Value Decomposition (SVD)	Improving recommender accuracy by focusing on diversity using explicit feedback

4.2 Precision@k

Setting "k"

- Is the user most likely interested in seeing top-N things in a recommendation system? That's why you should focus on the first N items when calculating precision and recall scores. Precision@k, which the user sets to match the top-N goals, are defined as the user-definable integer k.

Definition of precision

- The precision@k is the percentage of relevant features inside the top K set recommended mainly through the algorithm.
- The following is how it should be interpreted. For example, let's assume If our precision at 10 in a top-10 suggestion problem is 80 percent. This means that about 80% of the suggestions I make have some connection to the user's current circumstance.
- precision@k can be found below.

$$\boldsymbol{Precision} = \frac{\boldsymbol{No\ of\ our\ recommendations\ that\ are\ relevant}}{\boldsymbol{No\ of\ items\ we\ recommended}}$$

Table 4. 4: Precision@k Results Comparison

K	General CF Model	CDRS Model
10	0.813728207129846	0.834262945615404 ↑
20	0.813589004565505	0.824252557601032 ↑
30	0.799569140153965	0.819144608482268 ↑
40	0.796091551854567	0.814832328922597 ↑
50	0.788783975417775	0.812482904436214 ↑

This table shows that the CDRS model has the best precision@k performance under various k values. The proposed cross-domain recommender model's precision, recall, and F1 score are shown in the graph below. The CDRS model is expected to remain stable at K=30.

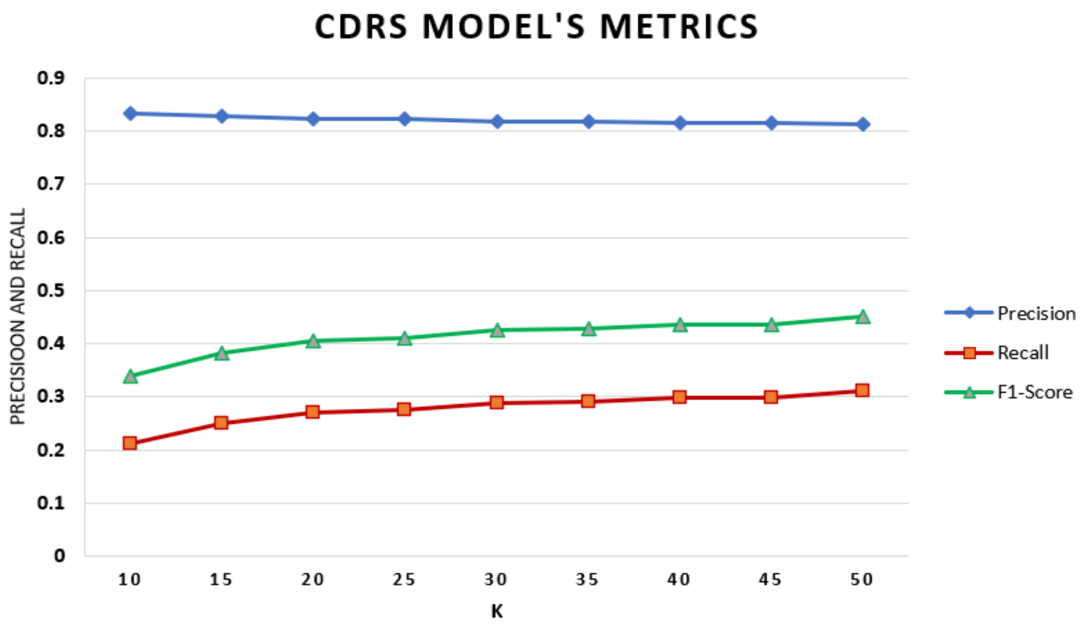

Figure 4. 9: CDRS -Precision, Recall & F1-Score

Here, below graph shows the result of precision values between some past research works. All models are focused on same data set.

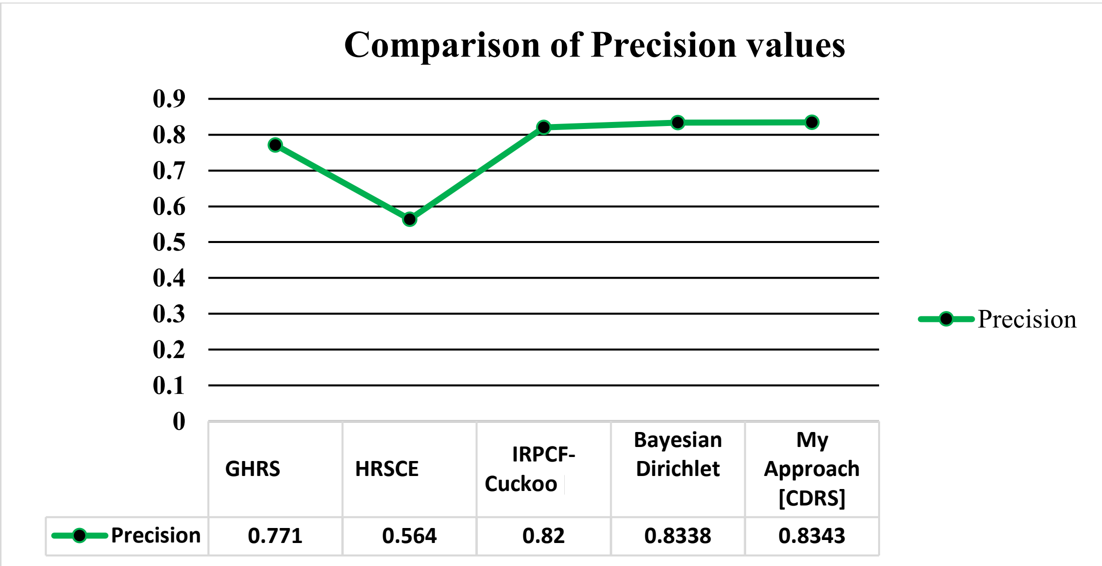

Figure 4. 10: Comparison of Precision values with different studies

By considering all above comparison results, it's clear that my proposed approach has showed an improvement in all matrices like RMSE, MSE, MAE and Precision.

4.3 Hit Rate

We can measure the recommender quality of top-N recommendations. We employ a hit rate to determine the top-10; If a user gives a positive rating to one of the top-10 we recommend, we consider it a "hit."

The procedure for calculating the hit rate for a single user is as follows:

- In the training data, locate all items in this user's history.
- Remove one of these items with the intent of doing so (Leave-One-Out cross-validation).
- Next, request the top 10 recommendations from the recommender using the remaining items.
- Once the removed item appeared in the top_N recommendation list, then it is considered a success. If it doesn't, it would not be a "hit".
- Here we can see out our calculated hit rate values are small since we are performing Leave-One-Out cross-validation.
- So, I have calculated the hit ratio in the same manner in both the general CF model and CDRS model and the results are shown in the below table.

Table 4. 5: Hit Ratio Results Comparison

Top N	General CF Model -Hit Ratio	CDRS Model-Hit Ratio
10	0.0365187188612096	0.0368655737704918 ↑
20	0.0550163934426229	0.0557137049180327 ↑
30	0.0678852459016393	0.0683718335583297 ↑
40	0.0786885245901639	0.0868852459016393 ↑
50	0.0918032786885245	0.0934426229508196 ↑

This table indicates the best Hit rate performances achieved by the CDRS model under Leave-One-Out cross-validation

4.4 Hit Rate by Rating value

- Predicted rating values can also be used to break down the hit rate. Our goal is to predict what people will like, so we worry about high ratings rather than low ones.
- I anticipated our hit rate to look like this, and it does, with a considerably greater hit rate for rating score 5 than for 4 or 3.
- Our CDRS model's hit rate by rating value is shown in the below table.

Table 4. 6: Hit Rate by Rating Value

Hit Rate by Rating value		
2.5	0.0366666666666667	↓
3.0	0.00869565217391301	↓
4.0	**0.0444444444444444**	↑
4.5	**0.0943396226415094**	↑
5.0	**0.0569105691056910**	↑

- Rating 5,4 has a substantially higher hit rate than the rest. As a result, the higher you go, the better.
- It's great that it has a high success rate for 4, 4.5, and 5-star ratings rather than low ratings.

Our proposed cross-domain recommender system can provide effective results for Rate by Rating value

Question: How we can Justify that proposed CDRS model has the best performance improvement?

I have made this question to show how my CDRS model has best performance rather than other recommender models.

From Table4.1: we can clearly see that,

Best values for the RMSE, MSE, MAE and FCP have been achieved by CDRS model. Even it has small difference in value vise but there is a RMSE, MSE, MAE & FCP improvement in CDRS --01

From Table4.4: we can clearly see that,
Best value for the Precision@k has been achieved by CDRS model --------------02

From Table4.5: we can clearly see that,
Best value for the Hit Ratio has been achieved by the CDRS model -------------03

Finally, according to all above 01,02 and 03 facts, it's very clear that my proposed CDRS has the best performance improvement via RMSE, MSE, MAE, FCP, Hit Rate and Precision@k evaluation protocols.

4.5 Diversity

- Although diversity appears to be a subjective measure, it can be determined by comparing the similarity of recommended items.
- We can compute the similarity between recommended items and average the similarity score.
- For the similarity matrix, similarity measure protocols such as cosine, Pearson, MSD, and Pearson_baseline can be applied.
- Compute the average similarity score of the top N recommendations (N).
- Diversity can be defined as the inverse of the average similarity score.
- Following formula can be used to calculate the diversity.

$$\boldsymbol{Diversity(Top@k) = 1 - Sim_k}$$

- Not only that I have calculated and measured diversity in my proposed CDRS model by using different kinds of similarity measures which are provided in surprise.

Table 4. 7: CDRS Diversity Results

Diversity in CDRS				Top N
Cosine	**MSD**	**Pearson**	**Pearson Baseline**	
0.031385722	0.44588575	0.776653201	0.964291724	10
0.035001745	0.44601719	0.787268406	0.970817127	20
0.037695422	0.445757136	0.792992469	0.974581553	30
0.040856006	0.446081011	0.797426866	0.977034585	40
0.044607557	0.447319215	0.800843425	0.978822341	50

- We can see when top-N increases then diversity also increases here.
- Highest Diversity value is achieved via Pearson & Pearson-baseline similarity measurements.

4.5.1 CDRS diversity via a Cosine similarity score

- Cosine Similarity score: Going to calculate the cosine similarity between all user pairs (or items)
- The "cosine similarity" is defined as like below

$$\text{cosine\_sim}(u, v) = \frac{\sum\limits_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum\limits_{i \in I_{uv}} r_{ui}^2} \cdot \sqrt{\sum\limits_{i \in I_{uv}} r_{vi}^2}}$$

Figure 4. 11: Cosine Similarity

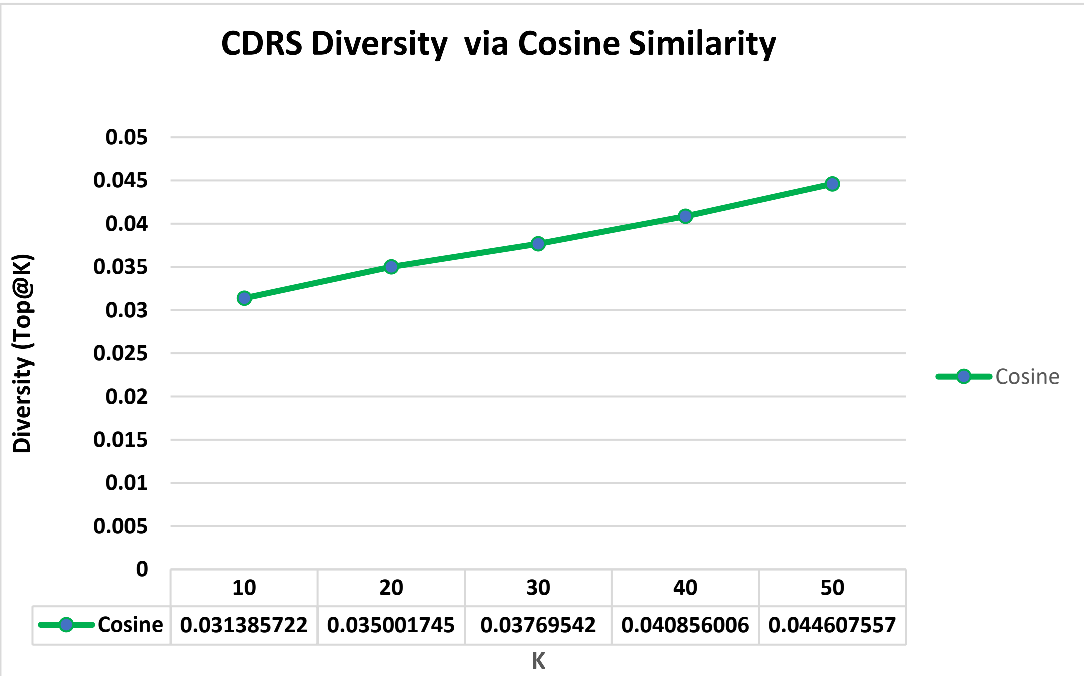

Figure 4. 12: CDRS Diversity via Cosine Similarity

4.5.2 CDRS diversity via MSD similarity score

- "MSD" similarity is defined as like below

MSD similarity score: Going to calculate Mean Squared Difference similarity between all pairs of users (or items)

$$\text{msd}(u, v) = \frac{1}{|I_{uv}|} \cdot \sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2$$

Figure 4. 13: MSD Similarity

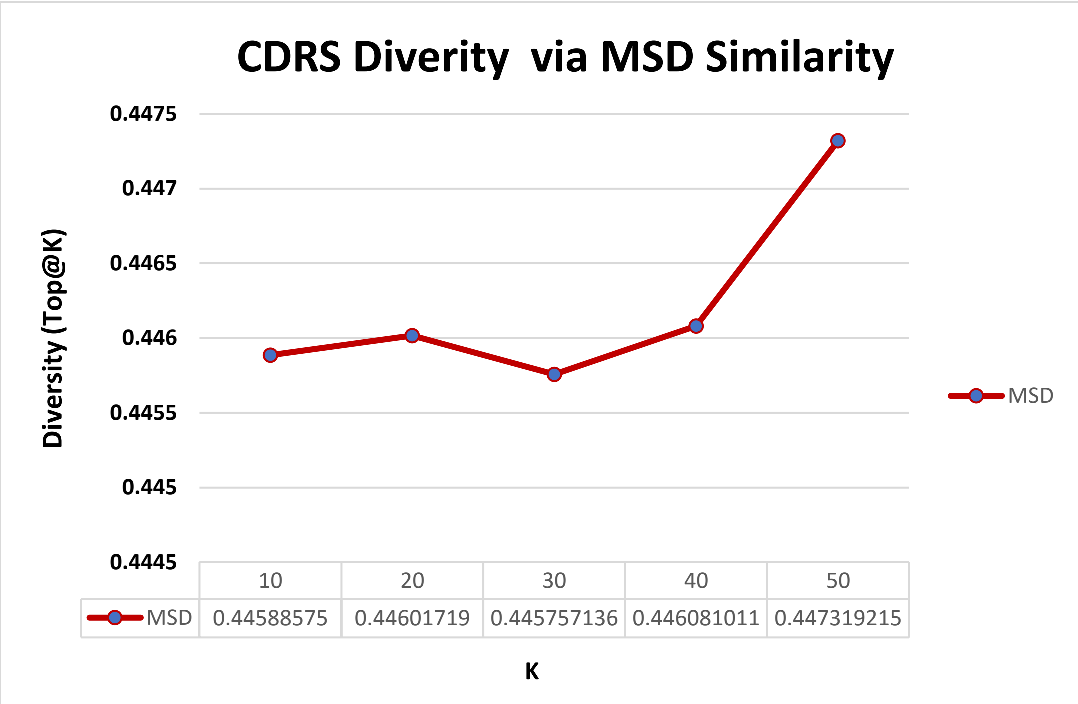

Figure 4. 14: CDRS Diversity via MSD Similarity

4.5.3 CDRS diversity via Pearson similarity score.

- Pearson-similarity score: For each user pair, we'll calculate Pearson's correlation coefficient (or items).
- It can be calculated from the below formula.

$$\text{pearson\_sim}(u, v) = \frac{\sum\limits_{i \in I_{uv}} (r_{ui} - \mu_u) \cdot (r_{vi} - \mu_v)}{\sqrt{\sum\limits_{i \in I_{uv}} (r_{ui} - \mu_u)^2} \cdot \sqrt{\sum\limits_{i \in I_{uv}} (r_{vi} - \mu_v)^2}}$$

Figure 4. 15: Pearson Similarity

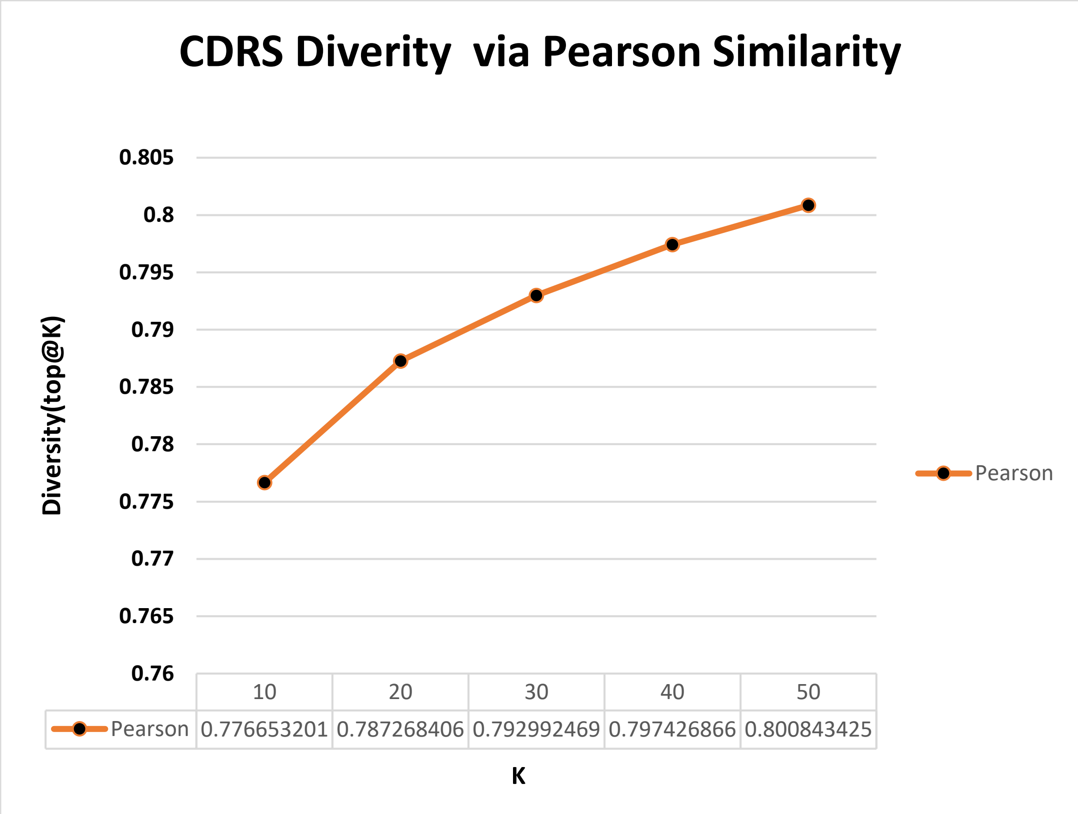

Figure 4. 16: CDRS Diversity via Pearson Similarity

4.5.4 CDRS diversity via Pearson_baseline similarity score.

- The Pearson_baseline similarity can be calculated from below equation.

$$\text{pearson\_baseline\_sim}(u, v) = \hat{\rho}_{uv} = \frac{\sum\limits_{i \in I_{uv}} (r_{ui} - b_{ui}) \cdot (r_{vi} - b_{vi})}{\sqrt{\sum\limits_{i \in I_{uv}} (r_{ui} - b_{ui})^2} \cdot \sqrt{\sum\limits_{i \in I_{uv}} (r_{vi} - b_{vi})^2}}$$

Figure 4. 17: Pearson Baseline Similarity

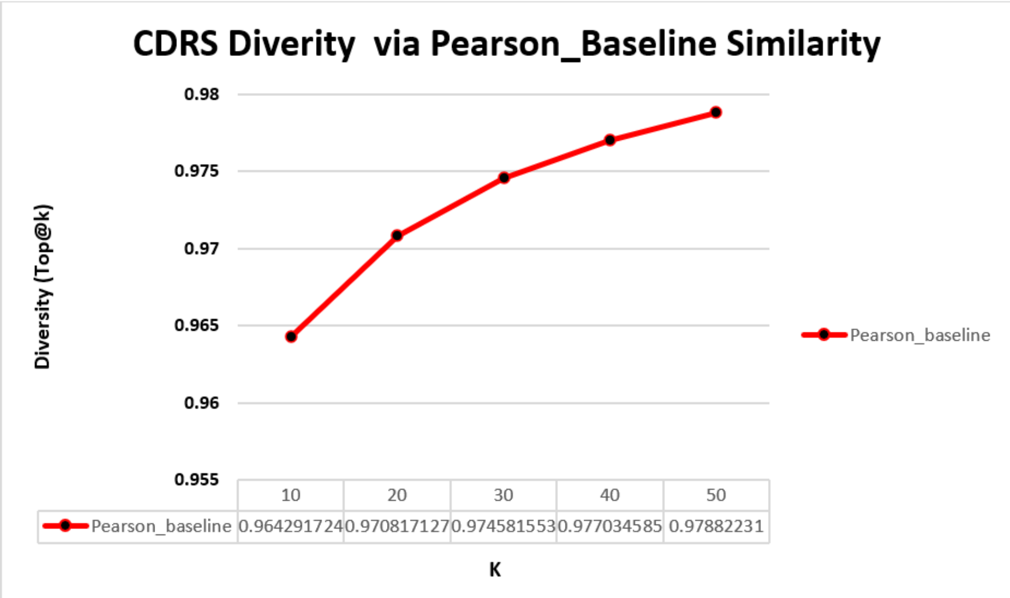

Figure 4. 18: CDRS Diversity via Pearson Baseline Similarity

4.6 Novelty

- The easiest method for quantifying novelty is the average popularity of top_N recommended items.
- We can measure novelty like how popular the items are that we are recommending.
- However, this is important because the point of recommender systems is to show items in what we call the "long tail," which is what we show.
- Once the K value is increased, novelty also increasing, and it’s clearly proved in below graph.

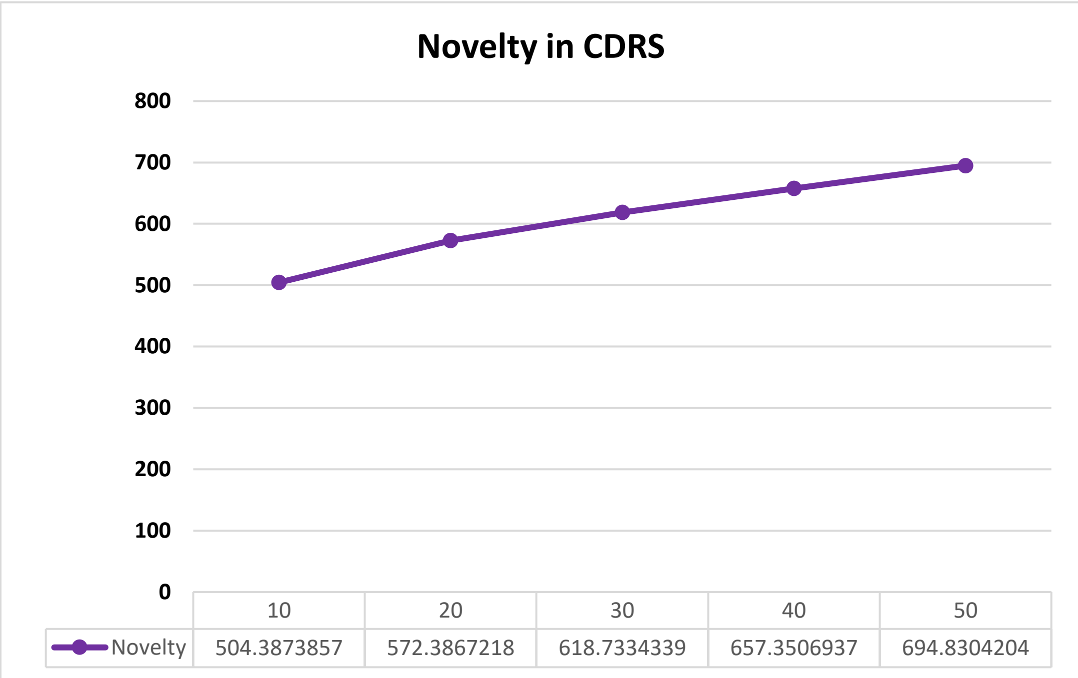

Figure 4. 19: Novelty in CDRS

5: CONCLUSION

Under the major domain RS, cross-domain recommender system is one of the not much-explored areas and it needs more research works and there existing many research gaps. The most preferred method when it comes to cross-domain recommender systems is to analyze, measure and evaluate their significance using widely used predicted accuracy metrics such as RMSE, MAE, and so on. Since the more need than accuracy to maintain high-quality recommendations, we need to pay attention to a few specific areas beyond accuracy like diversity and novelty.

With the purpose of finding an effective solution here, I have tried to build a matrix factorization-based collaborative filtering cross-domain recommender system using explicit user feedback with movilens100k research data set. The major target was implementing a CDRS model with improving accuracy by focusing on diversity. Here I have built two CF recommender models using the SVD MF method, one is the general CF recommender model and the other one is a CDRS model.

Basically, I could compare my CDRS model with the genera CF model. Furthermore, I have compared the proposed CDRS model's RMSE, MSE, MAE and precision performances with other recent recommender models which based on Movielense100K data as benchmarks. My CDRS model is a machine learning-based model and finally, it is achieved satisfactory RMSE, MSE, MAE and precision performance rather than other CF benchmark models. Not only that CDRS has good performance for MSE, MAE, and FCP accuracy protocols than the general CF model which I have built.

Furthermore, hit ratio results and Precision @k results are also far better achieved in CDRS rather than the general CF model. Those results clearly highlight that our proposed CDRS model has good recommender accuracy rather than a general collaborative recommender model.

Next, I calculated similarity between recommended items and got the average similarity score using four major similarity measure protocols like cosine, MSD, Pearson, and Pearson_baseline and CDRS model's diversity was calculated from all above for types. As per the CDRS diversity results, we could see when the Top-N recommender list is increasing model's diversity also increases. The novelty of the CDRS also calculated by me and that is also increased when top-N increases. Especially after the top-30 recommender list, diversity, and novelty both are increasing rapidly.

Finally, here we could achieve our expected goal and these findings would be pretty much worthy when we are implementing enterprise-level cross-domain recommender systems to reach modern business use cases as the future work we hope to expand this CDRS model via content-based filtering and a new hybrid approach to reach more recommender user goals.

REFERENCES

[1] R. B. a. Y. Koren, "“Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights”," *IEEE International Conference on Data Mining (ICDM’07),* pp. 43-52, 2007.

[2] J. K. G. K. S. John Riedl, "Item-based collaborative filtering recommendation algorithms," *WWW '01: Proceedings of the 10th international conference on World Wide Web,* no. 01 April 2001, pp. 285-295, 2001 MAY.

[3] M. d. G. &. G. S. Pasquale Lops, "Content-based Recommender Systems: State of the Art and Trends," in *Recommender Systems Handbook*, 2011, pp. 73-105.

[4] R. J. M. Loriene Roy, "Content-based book recommending using learning for text categorization," *DL '00: Proceedings of the fifth ACM conference on Digital libraries,* pp. 195-204, 2000.

[5] M. Z. A. F. Dietmar Jannach, Recommender Systems, Cambridge University Press, 2010.

[6] J. W. D. M. M. W. W. a. Z. G. Lu, "Recommender System Application Developments: A Survey. Decision Support Systems," *Journal of Software Engineering and Applications,* vol. 74, pp. 12-32, 2015.

[7] I. &. C. I. &. K. M. &. R. F. Fernández-Tobías, "Cross-domain recommender systems: A survey of the State of the Art," *Proceedings of the 2nd Spanish Conference on Information Retrieval,* 2012.

[8] V. R. Atisha Sachan, "A Survey on Recommender Systems based on Collaborative Filtering Technique," 2013.

[9] A. G. Lalita Sharma, "A Survey of Recommendation System: Research Challenges," *International Journal of Engineering Trends and Technology (IJETT),* vol. 4, no. 5.

[10] P. K. D. P. P. C. Pradeep Kumar Singh, "A Comparative Study of Different Similarity Metrics in Highly Sparse Rating Dataset," *2nd International Conference on Data Management, Analytics & Innovation (ICDMAI 2018),* vol. 2, 2018.

[11] P. P. N. D. P. C. PK Singh, " A Novel Neighborhood Calculation Method by Assessing Users’ Varying Preferences in Collaborative Filtering," *ISCA 34th International Conference on Computers and Their Applications (CATA 2019), March 2019, Honolulu, Hawaii, EPiC Series in Computing,* vol. 58, pp. 345-355, 2019.

[12] H. Y. H. T. Songjie Gong, "Combining Memory-Based and Model-Based Collaborative Filtering in Recommender System," *2009 Pacific-Asia Conference on Circuits, Communications and Systems,* 2009.

[13] F. &. R. L. &. S. B. Ricci, "Recommender Systems Handbook," *Recommender Systems Handbook,* Vols. 1-35, pp. 1-35, 2010.

[14] D. B. Michael J. Pazzani, "Content-Based Recommendation Systems," in *Content-Based Recommendation Systems*, 2007, pp. 325-341.

[15] R. B. Alexander Felfernig, "Constraint-based recommender systems: technologies and research issues.," in *10th International Conference on Electronic Commerce 2008, Innsbruck, Austria*, 2008.

[16] M. a. &. P.-B. Ghazanfar, "A Scalable, Accurate Hybrid Recommender System," Phuket, Thailand, 2010, pp. 94-98.

[17] P. B. T. M. G. Barve, "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System," *International Journal of Computer Applications (IJCA),* vol. 110, 2015.

[18] A. K. A. B. Anand Shanker Tewari, "Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining," in *2014 IEEE International Advance Computing Conference (IACC)*, 2014.

[19] G. Y. Z. C. Y. D. Yingjie Wang, "A trust-based probabilistic recommendation model for social networks," in *Trust Mechanisms for Crowdsourcing Systems in Mobile Social Networks*, 2015.

[20] W. S. H. J. P. Y. L. S. W. K. D. L. Jongwuk Lee, "L-Injection: Toward effective collaborative filtering using uninteresting items," in *JournalIEEE Transactions on Knowledge and Data Engineering*, 2019.

[21] Y. S. X. H. Ali Mamdouh Elkahky, "A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems," in *WWW '15: Proceedings of the 24th International Conference on World Wide Web*, 2015.

[22] P. C. J. A. E. Sargin, "Deep Neural Networks for YouTube Recommendations," in *10th ACM Conference on Recommender Systems, ACM, New York, NY, USA (2016)* , NY, 2016.

[23] H. L. D. C. S. L. P. G. &. J. G. Sheng Gao, "Cross-Domain Recommendation via Cluster-Level Latent Factor Model," *ECML PKDD 2013: Machine Learning and Knowledge Discovery in Databases ,* pp. 161-176, 2013.

[24] Q. Y. X. X. Bin Li, "Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction," in *IJCAI*, 2009.

[25] Z. L. &. H. H. Xin Xin, "A Nonlinear Cross-Site Transfer Learning Approach for Recommender Systems," *ICONIP 2014: Neural Information Processing,* pp. 495-502, 2014.

[26] B. C. D.-Y. Y. Yu Zhang, "Multi-Domain Collaborative Filtering," in *Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI2010)*, 2012.

[27] T. K. S. Berkovsky, "Cross-domain mediation in collaborative filtering. In User Modeling 2007," in *Knowledge Science, Engineering and Management: 10th International Conference*, 2007.

[28] E. Z. L. Z. E. W. X. W. P. a. Q. Y. Zhongqi Lu, "Selective Transfer Learning for Cross Domain Recommendation," *2013 SIAM International Conference on Data Mining (SDM),* pp. 641-649.

[29] S. J. P. Q. Y. Lili Zhao, "A unified framework of active transfer learning for cross-system recommendation," *Advancement of Artificial (www.aaai.org),* vol. 245, pp. 38-55, 2017.

[30] T. M. W. Z. S. L. Suman Deb Roy, "SocialTransfer: cross-domain transfer learning from social streams for media applications," in *20th ACM international conference on Multimedia*, 2012.

[31] Y. S. M. L. &. A. H. Babak Loni, "Cross-Domain Collaborative Filtering with Factorization Machines," *ECIR 2014: Advances in Information Retrieval ,* pp. 656-661, 2014.

[32] Y. K. a. C. V. Robert M. Bell, "The BellKor solution to the Netflix Prize," *AT&T Lab,* 2008.

[33] J. S. Yehuda Koren, "Collaborative filtering on ordinal user feedback," *Twenty-Third international joint conference on Artificial Intelligence,* pp. 3022-3026, 2013.