

RECOMMENDER SYSTEM FRAMEWORK BASED ON DATAMINING TECHNIQUES

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Master of Science in
Computer Engineering**

**by
Nevzat KAYA**

**June, 2011
İZMİR**

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**RECOMMENDER SYSTEM FRAMEWORK BASED ON DATAMINING TECHNIQUES**” completed by **NEVZAT KAYA** under supervision of **ASST. PROF. DR. DERYA BİRANT** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Asst. Prof. Dr. Derya Birant

Supervisor



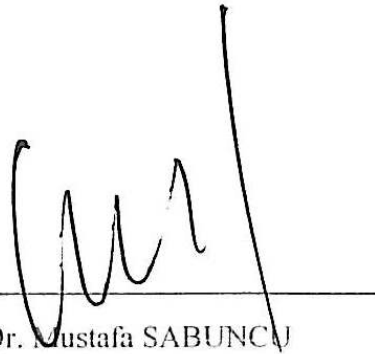
Prof. Dr. Alp Kurt

(Jury Member)



Asst. Prof. Dr. Özge ŞAHİN

(Jury Member)



Prof. Dr. Mustafa SABUNCU
Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to thank to my supervisor, Asst. Prof. Dr. Derya Birant, for her support, supervision, patience, understanding, kindness and useful suggestions throughout this study.

I would like to express my gratitude to my family and my friends for their support during the time of the project.

Nevzat KAYA

RECOMMENDER SYSTEM FRAMEWORK BASED ON DATA MINING TECHNIQUES

ABSTRACT

Product recommendation is a business activity that helps users to make the right decision and to decrease the time of waste and money. With increasing data amount, it is becoming popular day by day.

There are two approaches to implement the recommender system: Collaborative Filtering and Content Based Filtering. *Collaborative filtering* thinks that what a user thinks in the past thinks same now and in the future. It tries to find close users. *Content based filtering* deals with searches and clicks of the user. It recommends similar items to those items.

This thesis presents a framework of personalization expert by combining Collaborative Filtering method and Association Rule Mining technique. *Collaborative filtering* provides to reach the closest (similar) users for a user in point of taste. Users can view similar users more than the others and have an opportunity to examine and follow more. They can exchange their ideas among themselves. Each user affects the users that are close to him in point of prediction score. *Association rule mining* technique provides to find itemsets that have strong relationship between actual items. This technique is implemented by considering only the items that are liked by people. In addition, genre properties of the items are also considered in time of searching the items which users who like actual item like other else. The items which have strong relationships between actual items are listed. In this way, both prediction score and associated items that similar users liked are listed together. Users can take advantage of comments that are done related to the item.

In order to demonstrate the efficiency of proposed model, a movie recommender application, CinreC, was developed. The model was constructed independently from the type of item. The system can be converted to other recommendation systems, such as books, music, TV programs, trips, news recommender systems by only

changing the user interface. The experimental results show that proposed algorithm can efficiently perform online dynamic recommendation in a stable manner.

Keywords: Collaborative Filtering, Association Rule Mining, Model Based, Memory Based, Item Based, Pearson Correlation, User Based, Content Based Filtering

VERİ MADENCİLİĞİ TEKNİKLERİNE DAYALI TAVSİYE SİSTEMİ

ÖZ

Ürün tavsiyesi kullanıcıların doğru kararı vermesine yardımcı olan zaman ve para kaybını azaltan bir iş aktivitesidir. Artan veri miktarı ile birlikte günden güne popüler hale gelmiştir.

Tavsiye sistemini gerçeklemek için iki yaklaşım vardır: İçerik Tabanlı Filtreleme ve İşbirlikçi Filtreleme. *İşbirlikçi filtreleme* bir kullanıcı geçmişte ne yapıyorsa şimdi ve gelecekte de aynı şeyi yapacağını düşünür. Yakın kullanıcıları bulmaya çalışır. *İçerik tabanlı filtreleme* kullanıcının aramalarıyla, sayfa tıklamalarıyla ilgilenir. Bunlara benzer öğeleri tavsiye eder.

Bu tez, Birliktelik Kuralı Madenciliği ve İşbirlikçi Filtreleme methodunu kullanarak kişiye özel bir framework sunar. *İşbirlikçi filtreleme* zevk bakımından bir kullanıcıya en yakın (benzer) kullanıcıları bulmayı sağlar. Aktif kullanıcı bu kullanıcıları diğer kullanıcılardan daha fazla görür ve daha fazla inceleme ve takip etme imkanına sahip olur. Kendi aralarında fakir alış verişinde bulunabilirler. Her bir kullanıcı tavsiye puanı bakımından kendisine yakın olan kullanıcıları etkiler. *Birliktelik kuralı madenciliği* sayesinde listelenen öğeyle arasında güçlü bir ilişki olan diğer öğeler bulunmasını sağlar. Bu teknik kişiler tarafından sevilen öğeler göz önünde bulundurulur. Ayrıca bir öğenin başka hangi öğelerle beğenildiği araştırılırken öğenin tür özellikleri de göz önünde bulundurulur. Aralarında güçlü bir ilişki bulunan öğeler listelenir. Böylelikle tavsiye puanı ve bu öğeyi sevenlerin başka hangi öğeleri sevdikleri birlikte gösterilmiş olunur. Kullanıcılar öğeye ilişkin yapılan yorumdan faydalanabilir.

Önerilen modelin işleyişini göstermek için bir film tavsiye sistemi, CinreC, geliştirilmiştir. Model öğenin türünden bağımsız olarak geliştirildi. Sistem sadece ara yüz değiştirilerek kitap, müzik, TV programı, gezi, haberler gibi çeşitli tavsiye sistemlerine dönüştürülebilir. Deney sonuçları da önerilen algoritmanın istikrarlı ve etkili bir şekilde çalıştığını göstermektedir.

Anahtar Sözcükler: İşbirlikçi Filtreleme, Birliktelik Kuralı Madenciliği, Model Tabanlı, Hafıza Tabanlı, Öğe Tabanlı, Pearson Kolerasyon, Kullanıcı Tabanlı, İçerik Tabanlı Filtreleme

CONTENTS

	Page
M.Sc THESIS EXAMINATION RESULT FORM	ii
ACKNOWLEDGMENTS.....	iii
ABSTRACT	iv
ÖZ.....	vi
CHAPTER ONE-INTRODUCTION	1
1.1 Introduction.....	1
1.2 Motivation.....	2
1.3 The Purpose of Thesis	3
1.4 Thesis Organization.....	4
CHAPTER TWO – RECOMMENDER SYSTEM.....	5
2.1 Data Mining	5
2.2 Recommendation System	6
2.3 Recommendation System Methods	6
2.3.1 Collaborative Filtering	6
2.3.2 Content Based Filtering.....	13
2.3.3 Association Rule Mining	13
2.4 Related Work	15
CHAPTER THREE – MOVIE RECOMMENDER.....	16
3.1 Cinrec.....	16
3.1.1 CinreC Structure	17
3.2 Recommendation Methods	18
3.2.1 User Based Collaborative Filtering	18
3.2.2 Item-Based Collaborative Filtering.....	25
3.3 Database Structure.....	30
3.3.1 Stored Procedures	31
3.3.2 Views.....	35

3.4	Social Network.....	38
CHAPTER FOUR – APPLICATION AND RESULTS		40
4.1	Scenario.....	40
4.2	Data Set Description	42
4.3	Used Technologies and Programming Languages	42
4.4	User Interface	42
4.5	Experimental Results.....	56
4.6	Comparison with Other Systems.....	60
CHAPTER FIVE – CONCLUSION AND FUTURE		64
5.1	Conclusion	64
5.2	Future Work.....	65
REFERENCES		66

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Recommender systems are becoming more important over time. Recommender systems are generally used in e-commerce and in social networking sites. There are lots of advantages of them. In the systems where accuracy of recommended items has big percent, system users enjoy more in very short time. Recommender systems are guide that shows users' taste to users. This makes the recommender system popular. In this thesis, recommendation comes true in two ways. In the first one, users rank items, the system finds similar users and then it recommends to user according to another users. In the second one, the system finds similar items and then it recommends to user according to user's item choice and it finds similar items of that.

There are many methods for building a recommender system. However, one of the main challenges in recommender systems is that size of database increases fast. Users are continuously rating the items between a specific range such as between 0 and 5. Thus database is expanding so much every day, analysis time is increasing regularly. In this thesis, in order to provide best performance and best accuracy, all methods were examined, tested and compared and then the best one was chosen.

People, who do not know each other but their tastes are similar, meet in this system. For example, if he/she wants to watch a film, he/she can find the rating which is determined by the system according to analysis. Similarly, top recommendation list is listed and users decide the film which they will watch. Similarly, when he/she watched a film and liked so much, the recommender system can determine the list of films that he/she will possibly like.

Numbers of books, songs, movies are increasing every day. The company of them is making advert for selling more. Customers generally don't know anything about these new items and for this reason they have difficulties in choosing among these products. They only watch the adverts, and then they buy the products which they like the products' adverts. Another reason for buying is to like an actor of a film or a writer of a book. However, although this actor has a lot of good films or this writer has a lot of good books, and it is possible that the customer can't like this one, differently from the previous items.

1.2 Motivation

There are lots of films, books, songs in the world. The number of them is increasing every day. People do not know how to choose before they have watched/listened or read them. They ask to their friends, they read comments and they form a judgement about the item. But this is not enough. What is the relation between them? Some members of the system have similar tastes, but some members may have very different tastes from each other.

People have limited friends in some places and moreover, they have no friends in some cities or countries. Web sites allow them to find friends from all around the world. But many web sites do not say anything about other members; just show their basic information and their sharing, not any measurements between us and them.

Some people like a film and they watch it many times, they always watch this film because they aren't sure that another film is similar to this film. Recommender systems overcome this problem by finding similar items.

In the 21st century, time and money are important parameters; people have short time to enjoy, and so they should evaluate this time very well. For example, they watch only one film in one week properly and for this reason, they should choose this film carefully in a good way. Because some people have only three hours in one week for watching the movies. Otherwise three hours will be waste of time and they will not be able to enjoy this week because of bad chosen film.

Nearly 1.639.999 movies are registered in imdb.com site which has the largest movie database. Generally, each movie takes for 1.5 hour. All of them take for $1.639.999 * 1.5 = 2.459.998$ hours. It means approximately 280 years. Nobody lives for 280 years. Even someone lives for 280 years; nobody is busy with movies during all of his/her life.

In order to increase the enjoyable time and to decrease the waste of money, people need recommender systems. Because the life is short, and people should watch the movies earlier. In the world, six billion people live. People die without knowing most of them. In addition, a person should know closer people to meet them and to share some comments between themselves.

So, because of all these reasons, it is a good offer to recommend movies people like and also it is a good thing to find similar people to a person.

1.3 The Purpose of Thesis

Every year new books, films and songs are made. Also, many books, films, songs had been made and currently available. It is so difficult to choose something from these things. Because many people have not enough information except item names and stories. People may not decide which one they will buy.

The purpose of thesis is to help people decide which items they will choose and which items they will like, to predict their vote over the items and to list similar items to items that people liked. All these things provide more time and more enjoyable hours. CinreC also keeps like-minded users closer to share users' comments about films.

1.4 Thesis Organization

The thesis has five chapters. Chapter 1 presents project introduction, project motivation and the purpose of thesis. Chapter 2 describes the definition of recommendation systems, recommendation systems methods and currently available applications which were done before and how they work. Chapter 3 proposes a new hybrid model and explains the system theoretically. Chapter 4 explains how the system works and shows some experimental results. Last chapter presents future work and conclusion.

CHAPTER TWO

RECOMMENDER SYSTEM: STUDIES AND ISSUES

2.1 Data Mining

Data mining is a set of techniques that process the data and generate a result. These techniques provide to use the data sufficiently. They answer some questions. What can happen in the future? What should be done now? What is our complexion now? All these answers provide us to take precautions.

Technology has improved very fast. High size hard disks are produced. With this, it is not a problem to save all data. Most of the companies save all their data that is necessary or unnecessary in their applications. They only list everything or insert / store new data without doing data mining. Also internet in 21st century improved very fast. All this data is kept disorganized. Data mining helps them organized and useful. The useful information is provided to users and companies.

There are steps for implementing data mining. First of all, data is cleaned, all missing values are removed. All data change into particular form. The problem and the purpose are defined and the best technique of data mining techniques is determined for solving the problem.

Data mining is used on text data in web pages or systems which have a regular structure. Commonly used methods are:

- Association rule mining: It provides to find the relationship between the items and tries to search which items are connected with each other. This result is shown to user or is used for increasing selling. It can be summarized in the web sites 'people who liked this item, also liked these items'. In markets, items which have a relationship are put side by side for facilitating to be bought products or put the place where they are far from

each other to provide people to walk in the market more for buying more products.

- **Clustering:** It is an unsupervised learning problem which has not label. It makes a set of items which are similar to each other, another set of items which are different from the other sets of items. Number of set is determined by user. But the most appropriate 'number of set' can be created by examining the data. With text clustering, news can be categorized from internet. Same kind news can be put into same group. For example, news can read separately such as sport, magazine, politics etc.
- **Classification:** It is a supervised learning problem which has labels. Data is categorized to these labels. For example, the most popular example to guess that weather will happen tomorrow 'rainy', 'sunny' or 'cloudy'. Each term is a label.

2.2 Recommendation System

Recommender system is a system that presents to people some recommendations about items which people have not our idea about. Items can be books, films, songs, news, images etc. People who use a recommender system can watch, listen or read, after that they can have an idea about them. But it is important to have this idea before getting them. Recommendation systems supply this demand. It says sometimes "you will like these items", sometimes you will rate this item as a point over a point. In this way, items that the person likes can be closer than items that the person doesn't like.

There are a few approaches to implement recommendation system. Similarity between people can be found. This similarity can measure the rate people will give the rate for the items that they have no idea about and it can determine the films they will like. Similarity between items can be found. It can determine the relationship between items is strong or not. In this way, the item can be recommended to the person who liked an item that is strong relationship with this item.

When a person navigates in the web sites, they write some comments, search something and list some data more than other people. All this data gives information about this person. This data can be used for recommending the items. The things that the person always lists can be presented to this person at home page.

2.3 Recommendation System Methods

- Collaborative Filtering: This method searches similarity between users. It has two types:
 1. Model Based
 2. Memory Based
- Content Based Filtering: This method searches the user's web history, tries to collect information about user for recommending connected with that information (Lucue, 2010).
- Association Rule Mining: This method searches relationships between items. How much the relationships are strong is found.

2.3.1 Collaborative Filtering

Collaborative filtering is the technique of filtering data and can produce recommendations by computing similarity between the user's taste and the one of other users (Huang & Zeng, 2011). It makes a prediction based on weight of other users' opinions (Guo et al., 2009). It is also called social filtering (Para-Santander & Brusilovsky, 2010) (Guy & Carmel, 2011). It is the most common technique among the recommendations techniques. Collaborative filtering assumes that people do same things again that people did in the past. (Liu et al, 2010) So it determines like-minded users and it recommends to active user some items by using their rates they gave before and predicts the rates they will give. It is totally based on users.

There are three types of collaborative filtering.

2.3.1.1 Model Based

Model based collaborative filtering runs by using offline data. It determines a model. According to this model, recommends items (Kang et al, 2010). Clustering, Bayesian, principal component analysis are some of methods (Wikipedia). Jester (<http://eigentaste.berkeley.edu>) which used this system has a model. All of users rank same ten jokes (these jokes are chosen from different joke categories) and then the users are clustered. It recommends by using users' similarity. The disadvantages of this method; it runs offline and if it is thought film instead of joke, maybe a lot of people have not watched all of ten films. Reading joke takes one minute but watching film takes at least 1.5 hour. Because of these, in this thesis, this method is not used.

2.3.1.2 Memory Based

Memory-based algorithms use whole user-item database to generate a prediction. The data is always up-to-date. The processes are done online. A set of users are found and they called as neighbors who have sense like active user (Gong et al., 2009). The criterion is diversity on common items between the active user's ratings and the other users' ratings. It produces top n recommendations by utilizing these users (Khabbaz & LaksManan, 2011). Or it finds items' similarity. It says that users who liked this item also liked these items.

2.3.1.2.1 User Based. User based collaborative filtering generates active user's similar users. They called as neighbors too. Neighbors are found according to common items between active user and the other users. The similarity can be calculated by a few scaling methods.

$$S_{ab} = \frac{\sum_{i \in I_a \cap I_b} (v_{ai} - \bar{v}_a)(v_{bi} - \bar{v}_b)}{\sqrt{\sum_{i \in I_a \cap I_b} (v_{ai} - \bar{v}_a)^2 \sum_{i \in I_a \cap I_b} (v_{bi} - \bar{v}_b)^2}}$$

a, b	users
i	item
I_a	set items rated by user a
v_{ai}	rating of user a for item i
\bar{v}_a	average rating of user a

Figure 2.1 Pearson Correlation and Variables (Grčar et al., 2006)

Prediction is computed as shown in Figure 2.2 after finding the similarity distance between two users as shown in Figure 2.1.

$$p_{ai} = \bar{v}_a + \frac{\sum_{b \in U_i} s_{ab} (v_{bi} - \bar{v}_b)}{\sum_{b \in U_i} |s_{ab}|}$$

U_i	set users that have rated item i
-------	------------------------------------

Figure 2.2 Calculation of Prediction

User based collaborative filtering has some challenges although it is used widely. The most basic challenges are sparsity and scalability (Redpath et al., 2010).

- **Sparsity:** Every recommender system has a large database. For example, in movie recommendation systems, people rank 100,200 or 300 sometimes 10 films. But there are millions films in the movie database. Spacing can be so much .This may cause the recommendations are poor (Abdelwahab et al., 2009). More people rate, better the system can recommend.
- **Scalability:** The number of items and users grow continuously. With the millions of users and items, making computation becomes very difficult. The system slows down and listing recommendations becomes too late. The active user do not wait that much. Because of this, in this thesis new method is determined for finding similar users.

An example of user based is shown in Figure 2.3. The films that Ali, Nuray, Ayse liked are listed. It is seen from this figure that Ali and Nuray are more similar. They

have two common films which they liked. “My sassy girl” film can be recommended to Ali because of the similarity between Ali and Nuray. Because Ali has not watched it.

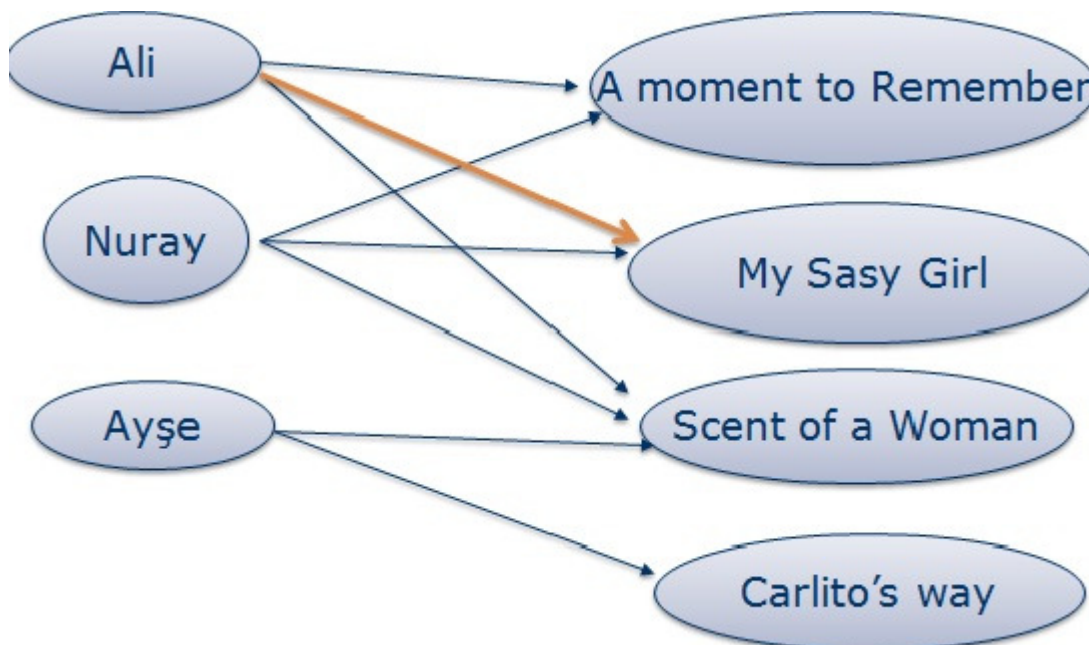


Figure 2.3 An example of User Based Collaborative Filtering

2.3.1.2.2 Item Based. If someone likes an item, system finds that item’s similar items (Gao et al, 2011). It is called item based collaborative filtering. In many of e-commerce sites, it is said that, “customers who bought this item also bought these”. They are listed below of it.

Some similarity measurements are (Sun et al., 2009):

2.3.1.2.2.1 Correlation-based Similarity. Similarity between two items i and j is measured by computing the Pearson-r correlation $corr_{i,j}$. the set of users who both ranked i and j are denoted by U then the correlation similarity is given by

$$sim(i, j) = corr_{i,j} = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}}$$

Here $R_{u,i}$ denotes the rating of user u on item i , \bar{R}_i is the average rating of the i -th item.

2.3.1.2.2.2 *Cosine-based Similarity*. In this case, two items are thought of as two vectors in the m dimensional user-space. The similarity between them is measured by computing the cosine of the angle between these two vectors. Formally, similarity between items i and j , denoted by $\text{sim}(i, j)$ is given by

$$\text{sim}(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

Where “ \cdot ” denotes the dot-product of the two vectors.

2.3.1.2.2.3 *Adjusted Cosine Similarity*. This similarity measurement is a modified form of vector-based similarity where it is taken into the fact that different users have different ratings schemes; in other words, some users might rate items highly in general, and others might give items lower ratings as a preference. To remove this drawback from vector-based similarity, average ratings are subtracted for each user from each user's rating for the pair of items in question: (Computer Science Comprehensive Exercise at Carleton College)

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}$$

An example of item based collaborative filtering is shown in Figure 2.4. All users liked “my sassy girl”. Ali and Nuray liked “a moment to remember” too. So, “a moment to remember” can be recommended to Ayse.

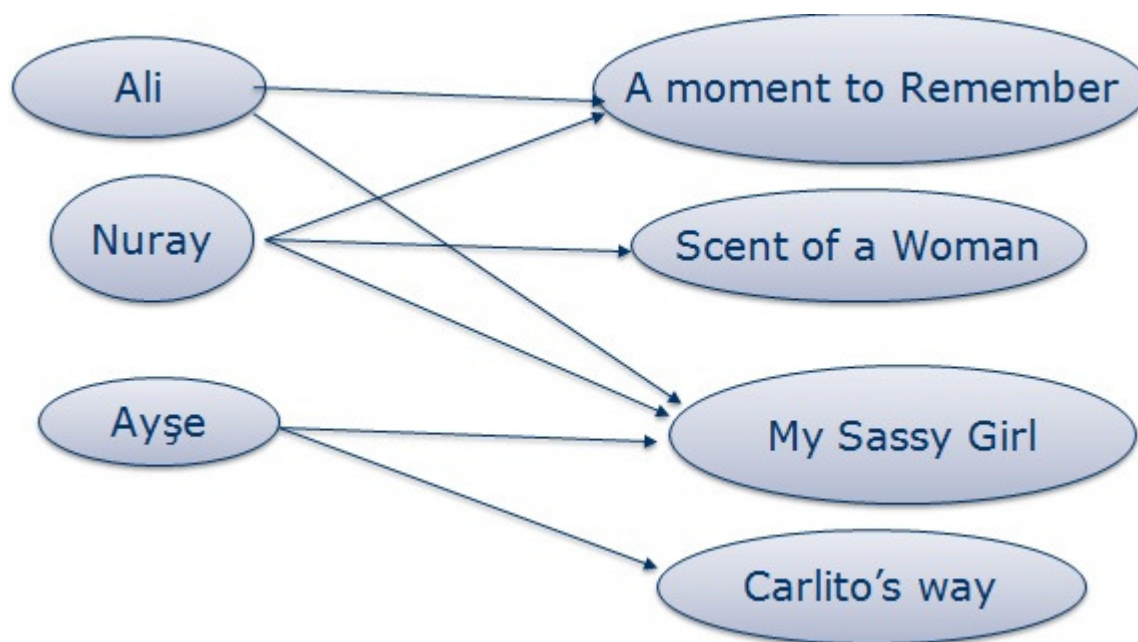


Figure 2.4 An example of Item Based Collaborative filtering

2.3.1.3 Hybrid

Hybrid approach combines a number of model based and memory based filtering algorithms (Kumar et al., 2010) (Liu et al., 2010). With this combination, it can overcome some limitations of collaborative filtering such as sparsity or slowing problem (Martinez et al., 2010).

2.3.2 Content Based Filtering

Content based filtering collects the contents of data. Often passing words, phrases, terms are used to build recommendation system. It is widely used in information retrieval. Some words pass often in the document. Thus, each of word has a vector expression. It is called weight. Similarity between two contents is measured according to their weights (Rohini & Ambati, 2006).

In social sites, users search something, list data, view pages, write comments. System would have an opinion what the user is interested in (Avancini et al., 2007) (Cai et al., 2010). Recommendation systems evaluate all this data. But content based

filtering cannot predict how much the user liked the item (Campos et al, 2010). For example it cannot say that it is four points over ten, just say you maybe like it.

2.3.3 Association Rule Mining

2.3.3.1 Description

Association rule mining is a method which finds interestingness measure or correlation between items in large databases. It is widely used in basket data analysis, cross marketing, catalog design and online-shopping web sites. It provides advantages to sell more. While the user is buying something, in that time user sees other products what other people bought this thing with. If some products are bought together, these products can be put to the different sides of market. In this way, the customer walks about all the market. He sees more things. In other words, there are more possibilities to buy more things.

2.3.3.2 Association Rule

Association rule is correlation between items based on some evaluation metrics. These metrics are determined by system designer. They are called support and confidence. With support, frequent itemsets are found. Finally when last versions of frequent itemsets are found, association rules are created. And these association rules must be ensured specific confidence.

TID(Transaction ID)	Items
1	City of angels, Rain man, Carlito's way
2	Carlito's way, Godfather I, Angela
3	Crown, Rain man, The city of angels, Godfather I

Figure 2.5 Transaction table of movies

Figure 2.5 shows that a movie is in a transaction or not. It is used for showing how to scale confidence and support below.

Rule is shown like that {the city of angels} \rightarrow {Crown}, {Godfather I} \rightarrow {Rain man}. Support and confidence scaling metrics determines that these rules are strong or not. If a rule is strong, there is a strong correlation between items.

Support is fraction of transactions that contain item set. Item set is a collection of one or more items. Support {the city of angels} = $2/3$ from Figure 2.5. Confidence: if it is shown that $c\{X \rightarrow Y\}$, it is said that confidence measures how often items in Y appear in transactions that contain X shown as Figure 2.6. Confidence {city of angels} \rightarrow {Crown} = $1/2$ from Figure 2.5.

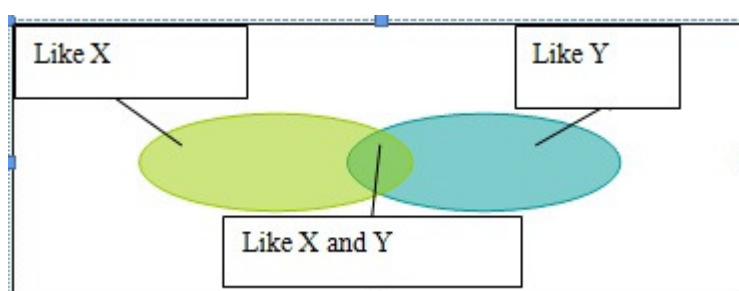


Figure 2.6 Place of support and confidence in a set.

Frequent item set: An item set whose support is greater than or equal to a minimum support.

2.4 Related Work

The recommendation systems grow up per day. They are now commonly used for product recommendation in e-commerce sites, market basket analysis or movie recommendation. There has been an international conference which is made by association for computer machinery (acm.org) since 2003 and the conference this year is in Barcelona on September 26-30. Practitioners and researchers come together and share their ideas.

The oldest recommendation system is Tapestry in 1992 (Goldberg et al., 1992). It was mail system to filter received documents. Users send their opinions about documents which were sent to user and Tapestry system sends e-mail according to

previous feedbacks. A user does not know similarity between him/her and other users. Because of that, Tapestry did not use collaborative filtering completely.

GroupLens.org (Resnick et al., 1994) is other recommendation system, recommends films to users. Users rate films. It uses user based collaborative filtering that uses Pearson correlation to find similarity between users. But Pearson correlation causes listing recommendations so slow. Ringo (Shardanand et al., 1995) is a music recommendation system. It is based on user based collaborative filtering too. It has some evaluation metrics instead of Pearson correlation for finding similarity between users.

Some clustering algorithms used in recommendation system. The performance of these algorithms was better than the nearest neighbor algorithms, but accuracy is smaller than them. In clustering for reducing dimensions, a method was used called as principal component analysis (Goldberg et al., 2001). Reducing dimensions was discussed in (Sarwar et al., 2000). Classification techniques are also used called as singular value decomposition with neural net classification (Billsus & Pazzani, 1998).

One of the biggest problems in user based collaborative filtering at first, when user enters to system was determined. Sparsity problem in dataset was mentioned (Sarwar et al, 1998) (Good et al., 1999). This sparsity problem is overcome by using item based collaborative filtering with association rule mining in this thesis.

CHAPTER THREE

MOVIE RECOMMENDER SYSTEM PROGRAMMING

3.1 CinreC

In this thesis, a movie recommender system called CinreC was developed. People register to system, rank movies and get some suggestions about movies. By using their rates, searches relationships between users and movies separately. If relationships are strong, it puts them closer and shows them to users primarily. To predict the rate for any film what they will score, it uses these relationships.

When system starts to work, hosting server pulls all data from database server. It is changed each data into object for working object oriented programming. Client sends users information and ratings of them, movie information to hosting server. Hosting server inserts this information to its cache and also sends to database server. All this data is processed in hosting server. When client sends request, information is sent by hosting server as shown in Figure 3.1.

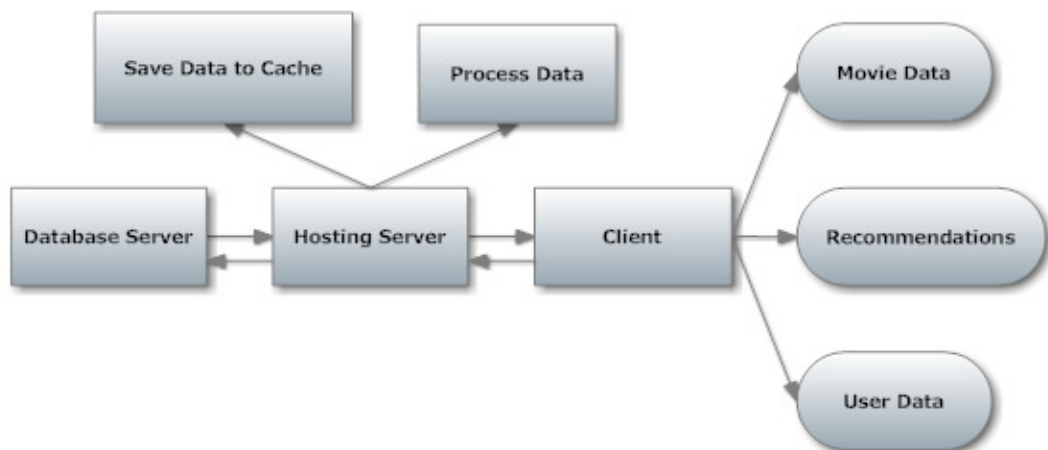


Figure 3.1 System Diagram

Figure 3.2 illustrates use case diagram of the recommendation system, what active user can do and what system does for the all users.

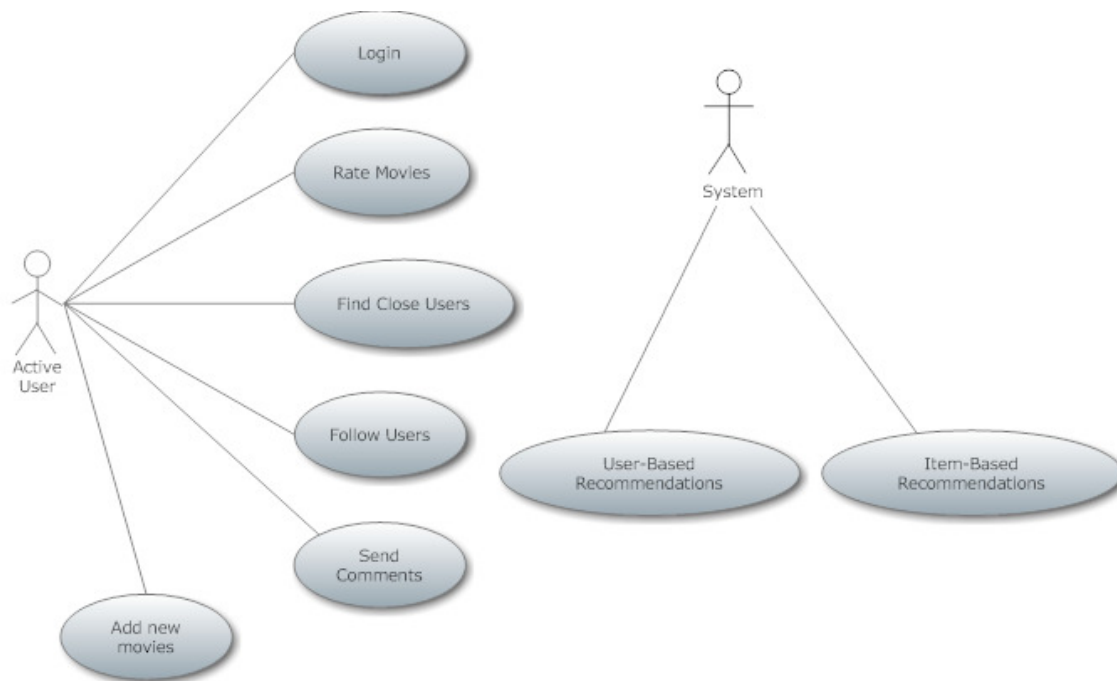


Figure 3.2 Use Case Diagram

3.1.1 CinreC Structure

The system is based on three-tier architecture. They are business, data access and user interface.

- **Data Access:** All database processes are done here. Object oriented structure is founded. After getting data, the data assign to objects. These objects are added to generic list. These generic lists are saved in static hash table. This process is done once. After that, if data is wanted, data is taken with static hash table. In only insert and update process, two operations happen. As first new data in database is updated or new data is inserted to database. As second hash table is updated or new data is inserted to hash table. The goals of these steps are to make contact with sql less. Thus, its performance increases and the recommendations are listed faster.
- **Business:** It only calls the methods in data access.
- **User interface:** The interfaces that users see are here. Getting recommendation page, generating new users, listing similar films are some pages in user interface. This tier contacts with business for getting data.

Some class diagrams are shown in figure 3.2. UserProcess class is used for finding common items between two users, similarity between two users. DbGetData is used for database operations.

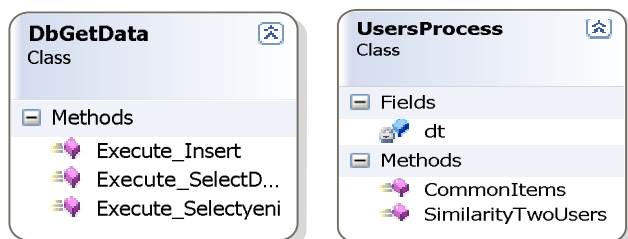


Figure 3.2 Classes and Methods

3.2 Recommendation Methods

Some of existing recommendation methods is used for predicting the rate which user will give and recommending the items. While recommendation methods are chosen, performance and accuracy of the predicted rate are considered.

3.2.1 User Based Collaborative Filtering

In our proposed model, user based collaborative filtering searches relationships dynamically between users by using history of active user. If this relationship is strong, it can be said these users are close to each other when the taste of movie is considered (Chen et al., 2010). Figure 3.3 shows how the system works generally. Rates of active user who enters the system are compared rates of other users and relationships are found. With these relationships, movies are recommended and predicted rates what active user will give.

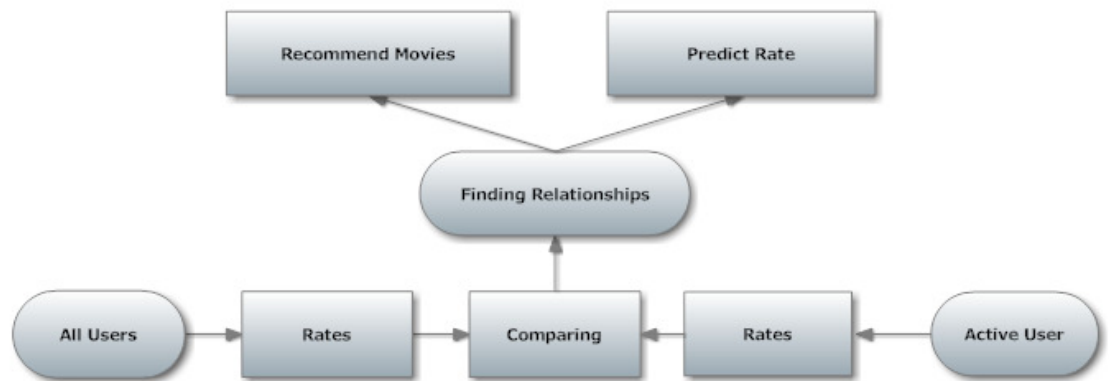


Figure 3.3 User Based Collaborative Filtering

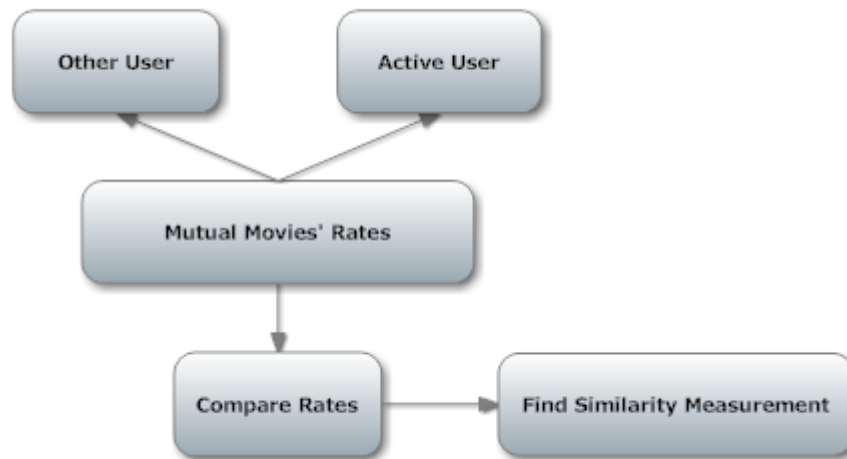


Figure 3.4 Finding Similarity Measurements between users

There are some measurements for finding similarity between users. Figure 3.4 illustrates how to scale similarity in the system. All measurements techniques so far are not enough for online recommendation systems. A technique is developed without damaging accuracy. Mutual films' rates belong to two people are subtracted and gotten absolute values of them. All these values are added and divided by number of mutual films. That is called similarity measurement. If similarity measurement is big, that user is not close to active user. The smaller it is, the closer that user is to active user. When a movie will be recommended or predicted the rate, these close users and their rates are used.

There are two functions to implement user-based collaborative filtering. They are written in C-Sharp programming language and they are called CommonItems and SimilarityUsers.

- CommonItems: finds mutual movies of two users. At first, it pulls movie data of the users from cache. It compares the movies and finds mutual ones.

```

public IList<TwoUserCommonItemsInfo> CommonItems(UserRateBr
kRbr, UserBr kbr, long userid1, long userid2, string con)
{
    ResultInfo user1;
    ResultInfo user2;
    IList<TwoUserCommonItemsInfo> LTuCItems = new
List<TwoUserCommonItemsInfo>();
    TwoUserCommonItemsInfo tcItems;
    IList<FilmInfo> lItems = new List<FilmInfo>();
    List<UserRateInfo> userList1 = new List<UserRateInfo>();
    List<UserRateInfo> userList2 = new List<UserRateInfo>();
    user1 = kRbr.DetailUser(userid1, con);
    userList1 = (List<UserRateInfo>)user1.data;
    user2 = kRbr.DetailUser(userid2, con);
    userList2 = (List<UserRateInfo>)user2.data;

    //Above the ratings of two users are pulling from the cache by ordering ids of
movies.
    if ((userList1 == null) || (userList2 == null))

```

```

        if ((userList1.Count == 0) || (userList2.Count == 0)) {
            return LTuCItems; }

    //Above if they have no common movies, it does not make any calculations. It
    can //not calculate similarity between them.

    if (userList1.Count > userList2.Count)
    {
        if(userList2[userList2.Count-1].Film.PrimaryKeyID
<userList1[0].Film.PrimaryKeyID)
            return LTuCItems;
        if(userList2[0].Film.PrimaryKeyID >userList1[userList1.Count-
1].Film.PrimaryKeyID)
            return LTuCItems;
        foreach (UserRateInfo ur in userList2)
        {
            if (ur.Film.PrimaryKeyID > userList1[userList1.Count -
1].Film.PrimaryKeyID)
                break;
            foreach (UserRateInfo ur2 in userList1)
            {
                if (ur.Film.PrimaryKeyID == ur2.Film.PrimaryKeyID)
                {
                    tcItems = new TwoUserCommonItemsInfo();
                    tcItems.Item = ur.Film;
                    tcItems.Rating1 = ur2.Rate;
                    tcItems = new TwoUserCommonItemsInfo();
                    tcItems.Item = ur.Film;
                    tcItems.Rating1 = ur.Rate;

```

```

        tcItems.Rating2 = ur2.Rate;
        LTuCItems.Add(tcItems);
        IItems.Add(ur.Film);
    }
    if (ur.Film.PrimaryKeyID < ur2.Film.PrimaryKeyID)
        break;
    }

    //Above if the second user has more movies than the second user, movie number
    of first user as upper limit is determined and added them to generic list by finding
    mutual movies.
    }
    }
    return LTuCItems;

    //Above the function returns generic list which has mutual movies of two users
    and their rates.
    }

```

- SimilarityUsers: finds similarity between two users by using mutual movies of two users. The rate of first user is subtracted from the rate of second user. Absolute of this value is taken and added to sum. This sum is divided by the number of mutual movies.

```

public SimilarityInfo SimilarityTwoUsers(UserRateBr kRbr, UserBr kbr, long
userid1, long userid2, string con)
{
    IList<TwoUserCommonItemsInfo>LTuCItems2=new
List<TwoUserCommonItemsInfo>();
    LTuCItems2.Clear();
    List<UserInfo> uTemp = new List<UserInfo>();
    UserInfo _kIInfo = new UserInfo();
    uTemp.Clear();
    ResultInfo rr = new ResultInfo();
    rr = kbr.Detail(userid1, con);

```

```

_kIInfo = (UserInfo)rr.data;
SqlConnection con2 = new SqlConnection(con);
uTemp.Clear();
rr = kbr.Detail(userid2, con);
_kIInfo = (UserInfo)rr.data;
SimilarityInfo sm = new SimilarityInfo();
sm.User = _kIInfo;
uTemp.Clear();
LTuCItems2 = CommonItems(kRbr,kbr, userid1, userid2, con);

```

//Above with ids of two users, objects of them are pulling from cache and sent to 'CommonItems' function for finding mutual movies. Necessary parameters are obtained for 'CommonItems' function.

```

double Result= 0;
double Similarity;
foreach (TwoUserCommonItemsInfo tuCItems in LTuCItems2)
{
    Result= Result+ Math.Abs(tuCItems.Rating1 - tuCItems.Rating2);
}

```

// Above the first user rate is subtracted from the second user rate and added to result.

```
sm.Total =Result;
```

```

Similarity= Result / LTuCItems2.Count;
sm.CommonItem = LTuCItems2.Count;
sm.Similarity = Similarity;
return sm;

```

//Above the result is divided by the number of mutual movies and found similarity between two users. The function returns similarity object which has similarity measurement and the number of mutual movies. }

So far the similarity between two users is calculated and it is time to calculate the recommendation score. A function for calculating it is developed. It is called 'PredictionScore'. It is shown below. Parameter in the function is set of users which score the active film. It is assumed that if the similarity between two users is zero distance, error coefficient is zero. If the similarity between the users 0.1 distance, error coefficient is 0.01. If the similarity distance increases, error coefficient also increases. It is thought that if two people have a distance, while the prediction score is being calculated, if the other person scores this film bigger than 2.5 point, this tolerance must be subtracted from the point; if the other person scores this film smaller than 2.5 point, this tolerance must be added to the point. It is thought if two people have a distance, this distance should affect in a reverse way from the user's score.

```

public double PredictionScore(DataTable dt10)
{
    i = 0;    Prediction= 0;
    for (int j = 0; j < dt10.Rows.Count; j++)
    {
        tempValue= Convert.ToDouble(dt10.Rows[j]["Rate"]) - 2.5;
        if (tempValue < 0)
        {
            Prediction+=(-tempValue)*
(Convert.ToDouble(dt10.Rows[j]["Similarity"])/10)+
Convert.ToDouble(dt10.Rows[j]["Rate"]);
        }
        if (tempValue > 0)
        {
            Prediction=Prediction-
(tempValue*(Convert.ToDouble(dt10.Rows[j]["Similarity"])/10))+Convert.ToDoub
le(dt10.Rows[j]["Rate"]);
        }
        if (tempValue == 0)
        {

```

```

        Prediction += Convert.ToDouble(dt10.Rows[j]["Rate"]);
    }
    i++;
}
Prediction= Prediction/ i;
return Prediction;
}

```

3.2.2 Item-Based Collaborative Filtering

It searches relationships between movies. When a movie is listed, it provides to be said ‘people who liked this movie, also liked these’. As shown in figure 3.5, the rates of movie that is listed which all users give and the rates of others movies which all users give are compared each other. If the rate of a movie is bigger than certain number, it is considered to compare active movie and this movie.

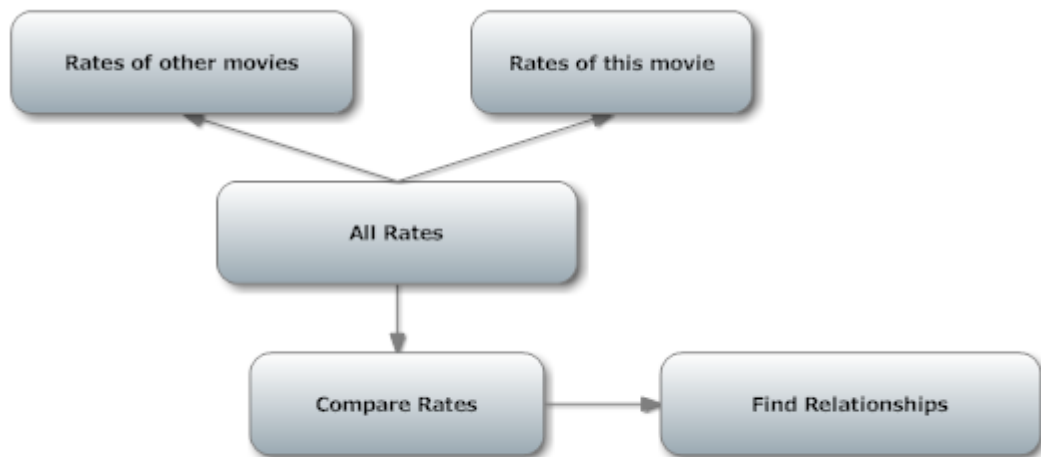


Figure 3.5 Item-Based Collaborative Filtering

Association rule mining is used for implementing item based collaborative filtering. Association rule mining searches relationships between items. With apriori algorithm, it is implemented with good performance.

Instead of listing all possible association rules, it says that if an itemset is frequent, its subsets must also be frequent, it generates from these subsets.



Figure 3.6 An example of Apriori Algorithm

Figure 3.6 shows how apriori algorithm works. Candidate itemset is composed and from this set, items that have support \geq minimum support are chosen. Frequent itemset is composed. This loop goes on until new candidate itemset can not be composed.

The goal is to find all rules having support \geq minimum support threshold and confidence \geq minimum confidence threshold (Cai et al., 2009).

Pseudo code of Apriori Algorithm:

```

L1= { frequent items };
for (k= 2; Lk-1 != $\emptyset$ ; k++) do begin
Ck= candidates generated from Lk-1 (that is: Cartesian product Lk-1 x Lk-1 and
eliminating any k-1 size itemset that is not frequent);
for each transaction t in database do
Increment the count of all candidates in Ck that are contained in t
Lk = candidates in Ck with minimum_support
end
return  $\cup_k$  Lk;

```

In apriori algorithm first of all, frequent itemset that has a one element is found. In a number of set of frequent itemset loop, candidate itemset is generated and eliminated ones which are not frequent. It finally finds final Lk (Mikoaj et al., 2004).

Data is gotten into generic list for finding frequent itemset. For decreasing to go to sql server, generic list is used. All generic lists are saved in static hash table when web site starts to work; in that time generic list comes to be ready. The queries are sent to this generic list. To find frequent itemset needs a lot of queries. Support of each itemset is found with these queries. If amount of support is bigger than the specific support amount, that item is added to list. The code of this is shown below.

```

public List<ItemCPlaceInfo> FrequentItems(List<ItemCPlaceInfo> _litemPlace, int
level,int supportCount)
{
    List<ItemCPlaceInfo> _lReturnItemC = new List<ItemCPlaceInfo>();
    int common_item = level - 2;
    string[] stringArray;
    string[] stringArray2;
    char[] seps = { '-' };
    int control = 0;
    string itemm = "";
    for (int m = 0; m < _litemPlace.Count; m++)
    {
        for (int k = m + 1; k < _litemPlace.Count; k++)
        {
            _icPlace = new ItemCPlaceInfo();
            _lBPlace = new List<LocationInfo>();
            control = 0;
            stringArray = _litemPlace[m].ItemSet.Split(seps);
            stringArray2 = _litemPlace[k].ItemSet.Split(seps);
            if (Convert.ToInt32(stringArray[0]) <
Convert.ToInt32(stringArray2[0]))
            {
                break;
            }
            for (int beginning = 0; beginning < common_item; beginning++)
            {
                if (stringArray[beginning] != stringArray2[beginning])
                { control = 1; break; }
            }
            if (control == 0)
            {

```

```

itemm = "";
for (int nn = 0; nn < stringArray.Length; nn++)
{
    if (itemm != "")
    {
        itemm = itemm + "-" + stringArray[nn];
    }
    else
    {
        itemm = stringArray[nn];
    }
}
for (int lk = common_item; lk < stringArray2.Length; lk++)
{
    itemm = itemm + "-" + stringArray2[lk];
    _itemc = new ItemCPlaceInfo();
    _itemc = (ItemCPlaceInfo)Detail(stringArray2[lk]);
}
amount = 0;
if (_litemPlace[m].Location.Count < _itemc.Location.Count)
{
    for (int n = 0; n < _litemPlace[m].Location.Count; n++)
    {
        for (int z = 0; z < _itemc.Location.Count; z++)
        {
            if(_litemPlace[m].Location[n].Place
            _itemc.Location[z].Place) <
            {
                break;
            }
        }
    }
    if(_litemPlace[m].Location[n].Place==_itemc.Location[z].Place)
    {
        amount = amount + 1;
        _bPlace = new LocationInfo();
        _bPlace.Place = _litemPlace[m].Location[n].Place;
        _lBPlace.Add(_bPlace);
    }
}
}
else
{
    for (int n = 0; n < _itemc.Location.Count; n++)
    {
        for (int z = 0; z < _litemPlace[m].Location.Count; z++)
        {

```

```

        if(_itemc.Location[n].Place
        _litemPlace[m].Location[z].Place)
        {
            break;
        }

    if(_litemPlace[m].Location[z].Place==_itemc.Location[n].Place)
        {
            amount = amount + 1;
            _bPlace = new LocationInfo();
            _bPlace.Place = _litemPlace[m].Location[z].Place;
            _IBPlace.Add(_bPlace);
        }
    }
}
}
}
if (amount >= supportCount)
{
    _icPlace.Counting = amount;
    _icPlace.Location = _IBPlace;
    _icPlace.ItemSet = itemm;
    _IReturnItemC.Add(_icPlace);
}
}
}
return _IReturnItemC;
}
}

```

3.3 Database Structure

User information, movie information, rates of users and some other details are saved in database. It is composed of nine tables. Figure 3.7 illustrates the relationships between tables.

- **Movies:** saves general information of movies. It is composed of name, subject, year, picture, imdb (internet movie database) link of movie.
- **User:** saves basic information of users. It is composed of user name, password, name, surname, job, e-mail, age, birth date, gender of user.
- **MovieRate:** saves users' ratings which they gave to the movies. It is composed of user, movie, rate and comment.

- **MovieGenres:** saves all genres of movies uniquely. It is composed of name of genre.
- **MovieMovieGenres:** saves the genre of each movie. It is composed of movie and its genre.
- **UserFriends:** saves friends of user. It is composed of user and his friend.
- **UserSimilarities:** saves similarity measurements between users. It is composed of active user, other user, number of mutual movie and similarity measurement.
- **FamousPeople:** saves actors, directors, scenarists who acted in the movies.
- **MoviePersonRelationship:** saves which famous person in which movie acted, directed.
- **Transactions:** saves relationships between movies. It is composed of movies and users who liked these movies and the number how many users liked these movies together.

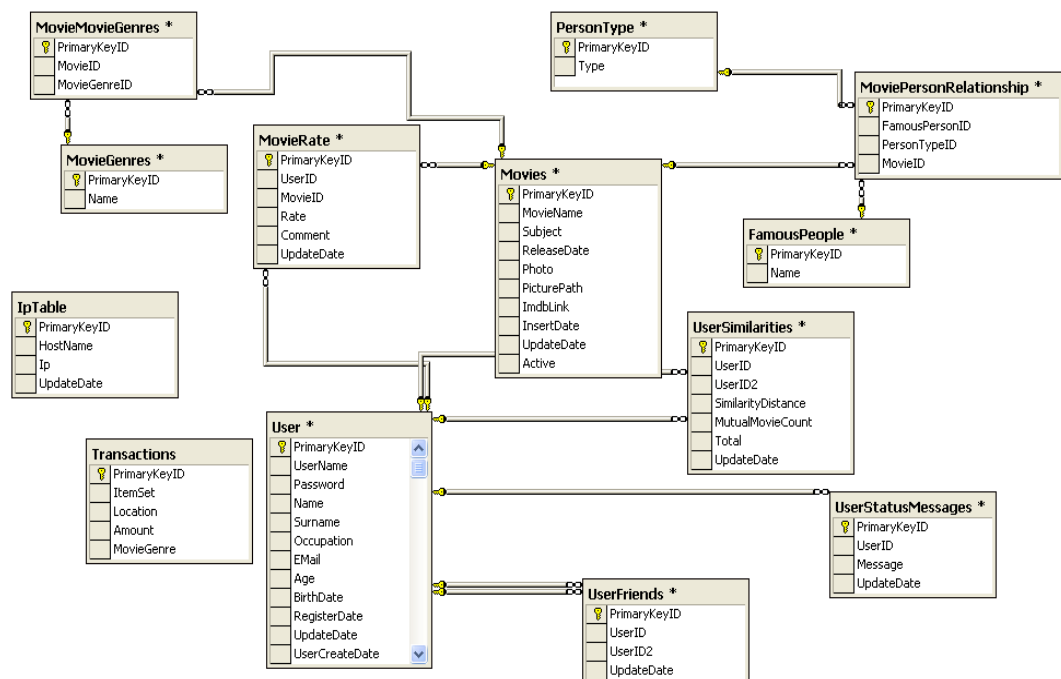


Figure 3.7 Database Structure

3.3.1 Stored Procedures

Stored procedures are used for updating, inserting, deleting, selecting data. Figure 3.8 shows which stored procedures are used. Most of them are basic stored procedures of the system. But some of them are very important for running the recommendation system.

- **NewTop75**: pulls seventy five users for calculating similarity measure between them and active user. It contains also users whose similarity was already calculated. If a user gives a score to a movie, if this was not calculated before, it is calculated now. The user who gave score most, they have priority. It takes user id as parameter. The stored procedure is shown below.

```
ALTER PROCEDURE [dbo].[NewTop75]
@ID as bigint
AS
SELECT top 75 *
FROM User
WHERE PrimaryKeyID
NOT IN( SELECT UserSimilarites.UserID2 AS PrimaryKeyID
FROM UserSimilarities
INNER JOIN
User ON
User.PrimaryKeyID = UserSimilarities.UserID2 INNER JOIN
User AS User_1
ON UserSimilarities.UserID2 = User_1.PrimaryKeyID AND
User_1.UpdateDate <= UserSimilarities.Date
WHERE (UserSimilarities.UserID = @ID))
and PrimaryKeyID NOT IN(
SELECT UserSimilarities_1.UserID AS SIRANO
FROM UserSimilarities AS UserSimilarities_1
INNER JOIN
User AS User_2 ON
```



```

User_2.PrimaryKeyID= UserSimilarities_1.UserID
INNER JOIN
User AS User_1 ON
UserSimilarities_1.UserID = User_1.PrimaryKeyID AND
User_1.UpdateDate <= UserSimilarities_1.Date
WHERE (UserSimilarities_1.UserID2 = @ID))
and PrimaryKeyID<>@ID
ORDER BY MovieCount DESC

```

- LastSimilarUsers2: shows like-minded users to active user in a sequential order. It takes user id as parameter. Stored procedure of it is shown below.

```

ALTER PROCEDURE [dbo].[LastSimilarUsers2]
@UserID as bigint
AS
(SELECT (SELECT COUNT(DISTINCT SimilarityMeasure)
FROM UserSimilarities
WHERE SimilarityMeasure <= t1.SimilarityMeasure
AND (UserID2 =@UserID or UserID=@UserID)
AND MutualMovieCount >= 5) AS OrderNumber,
t1.SimilarityMeasure,
t1.UserID2 as PrimaryKeyID,
User.Age, User.Occupation,
User.MutualMovieCount,
User.Gender,
t1.UserID,
t1.MutualMovieCount
FROM UserSimilarities t1
INNER JOIN
User ON t1.UserID2 = User.PrimaryKeyID
WHERE (t1.UserID = @UserID)
AND (t1.MutualMovieCount >= 5)
UNION
SELECT (SELECT COUNT(DISTINCT SimilarityMeasure)

```

```

FROM      UserSimilarities
WHERE     SimilarityMeasure <= t1.SimilarityMeasure
AND (UserID2 =@UserID or UserID=@UserID)
AND MutualMovieCount >= 5) AS OrderNumber,
t1.SimilarityMeasure,
t1.UserID as PrimaryKeyID,
User.Age,
User.Occupation, User.MovieCount,
User.Gender, t1.UserID2,t1.MutualMovieCount
FROM      UserSimilarities t1 INNER JOIN
User ON t1.UserID = User.PrimaryKeyID WHERE (t1.UserID2
= @UserID) AND (t1.MutualMovieCount >= 5))
order by OrderNumber asc

```

- UserRatesForPrediction: shows rates of ten users who are the nearest to active user in a sequential order. It takes movie id and user id as parameters. These ten users are people who gave a score to active movie. It is possible that they are not nearest users to active user. It is possible that some of nearest users did not give score to active movie. Sometimes ten users can be found. Because there are not ten users who gave a score to this movie. It takes user id and movie id as parameter. Stored procedure of it is shown below.

```

ALTER PROCEDURE [dbo].[UserRatesForPrediction]
@UserID as bigint,
@MovieID as bigint
AS
Select Top 10 SimilarityMeasure, Rate, UserID
From (SELECT
t1.SimilarityMeasure,
MovieRate.Rate AS Rate,
MovieRate.UserID AS UserID
FROM      UserSimilarities t1

```

```

INNER JOIN
MovieRate ON t1.UserID2 = MovieRate.UserID
WHERE ((t1.UserID = @UserID))
AND
(MovieRate.MovieID = @MovieID)
AND
(t1.MutualMovieCount>5)

UNION
SELECT
t1.SimilarityMeasure,
MovieRate.Rate AS Rate,
MovieRate.UserID AS UserID
FROM UserSimilarities t1
INNER JOIN
MovieRate ON t1.UserID = MovieRate.UserID
WHERE ((t1.UserID2 = @UserID))
AND (MovieRate.MovieID = @MovieID)
AND (t1.MutualMovieCount>5)
) as X
order by SimilarityMeasure

```

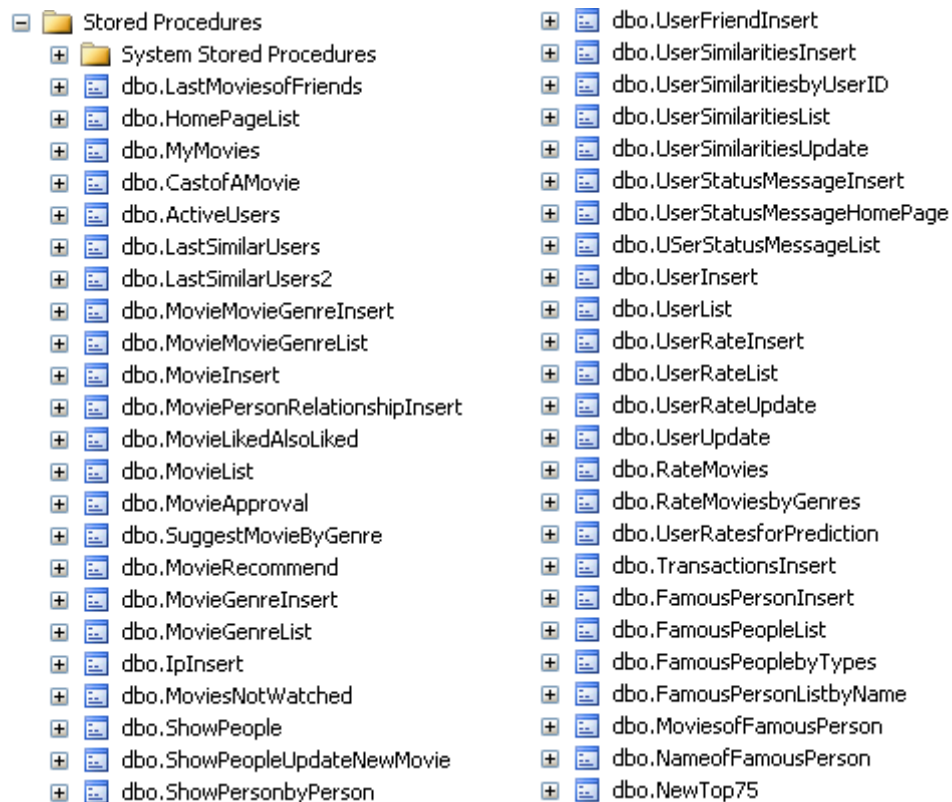


Figure 3.8 Stored Procedures

3.3.2 Views

View is virtual table that is composed of rows in other tables. A sql statement is written. With this statement, the data that is wanted is selected. Sql server does not need to resolve the query again and again. This increases the performance of the system. In this virtual table data is always up-to-date. Some views are used in CinreC to increase the performance.

- **MovieCount:** This view generates the count of each movie. In another words, it is found here how many times are given a score to each film. The structure of it is like that.

```

SELECT TOP (100) PERCENT COUNT(dbo.MovieRate.MovieID) AS
Score,
        dbo.MovieRate.MovieID
FROM    dbo.MovieRate INNER JOIN
        dbo.Movies ON dbo.MovieRate.MovieID =
dbo.Movies.PrimaryKeyID
GROUP BY  dbo.MovieRate.MovieID
ORDER BY COUNT(*) DESC

```

- **UserAverage:** This view generates the rating count of each user. It is found here how many times each user gave score. The structure of it is like that.

```

SELECT TOP (100) PERCENT AVG(dbo.MovieRate.Rate) AS Rate,
        dbo.MovieRate.UserID
FROM    dbo.MovieRate INNER JOIN  dbo.User ON
        dbo.MovieRate.UserID = dbo.User.PrimaryKeyID
GROUP BY  dbo.MovieRate.UserID
ORDER BY  dbo.MovieRate.UserID

```

- **MovieForPrediction:** This view is used for pulling data of user's closest friends to recommend. Which movies do closest friends of the active user like? This is found in this virtual table with this view. The structure of it is like this.

```

SELECT DISTINCT  dbo.Movies.PrimaryKeyID, dbo.Movies.Summary,
                dbo.Movies.ReleaseDate, dbo.Movies.PicturePath,
                dbo.Movies.MovieName, dbo.Movies.ImdbLink,
                dbo.UserSimilarities.UserID
FROM    dbo.MovieRate INNER JOIN  dbo.Movies ON
        dbo.MovieRate.MovieID = dbo.Movies.PrimaryKeyID INNER
JOIN
        dbo.UserSimilarities ON  dbo.MovieRate.UserID =
        dbo.UserSimilarities.UserID2
WHERE   (dbo.MovieRate.Rate >= 3) AND

```

```

        (dbo.UserSimilarities.MutualMovieCount >= 5) AND
        (dbo.MovieRate.Rate >= 3)
GROUP BY dbo.UserSimilarities.UserID,
        dbo.Movies.PrimaryKeyID,
        dbo.Movies.Summary,
        dbo.Movies.ReleaseDate,
        dbo.Movies.PicturePath,
        dbo.Movies.MovieName,
        dbo.Movies.ImdbLink
UNION SELECT  DISTINCT
        Movies_1.PrimaryKeyID,Movies_1.Summary,
        Movies_1.PicturePath,
        Movies_1.MovieName,
        Movies_1.ImdbLink,
        UserSimilarities_1.UserID2
FROM  dbo.MovieRate AS MovieRate_1
        INNER JOIN
        dbo.Movies AS Movies_1 ON
        MovieRate_1.MovieID = Movies_1.PrimaryKeyID
        INNER JOIN
        dbo.UserSimilarities AS UserSimilarities_1 ON
        MovieRate_1.UserID = UserSimilarities_1.UserID
WHERE  (MovieRate_1.Rate >= 3) AND
        (UserSimilarities_1.MutualMovieCount >= 5)
        AND (MovieRate_1.Rate >= 3)
GROUP BY
        UserSimilarities_1.UserID2,Movies_1.PrimaryKeyID,
        Movies_1.Summary,
        Movies_1.ReleaseDate,
        Movies_1.PicturePath,Movies_1.MovieName, Movies_1.ImdbLink

```

3.4 Social Network

Social network is the network that constructs relationships between people. It provides that people who know or don't know each other can communicate between each other. They share some comments, pictures, videos etc. In this way, they inform to each other about anything. People are met from another countries and cities. Maybe in real world there are not opportunities to meet them. But in the social networks, there are opportunities for this. But it is very difficult to meet, because if this person is not known in the real world. There is no reason to meet. Social networks do not provide any reason for people. Every person has some specific properties. If the system does not analyze this, mutual properties can't be known. But first of all, something should be presented to people to search mutual properties. If more things about people are known, the relationships between them easily can be found.

Movie recommender system, CinreC, is built in this thesis. Closest people to every person are found. It is thought they are found and only movies are recommended. Why doesn't it provide a social network with recommender system? People like this. People who the active user will like are found. Why don't they share their opinions between them? Why don't they watch any movie together? In this way, people will spend more time in the web site.

In other social networks, people who are known most are added to friend list. But this can be useless sometimes. People who are known may not express our feelings. They may think different from each other. Because of this, a short time later, people think there is nothing else to do here. They think that they spend time unduly. In social networks, people are chosen in these systems, system can not recommend anybody. These people can change and again these systems do not say anything. Always people must control their friends. But it takes effort and it really can cause to waste of time. In CinreC, these relationships change when people change. It does everything instead of the person. So, to add social network to CinreC will be useful in making good friendships and having good time more.

CinreC presents the advantage of social network and recommender system together. While users are getting suggestions, at the same time users can add a friend to his friend list, follow them what films they watched last.

Users also write comments about everything. These comments which are done by someone from user friend list are seen by active user. In this way, users keep in touch with like-minded users or others who active user chooses and anymore a reason exists for being a friend.

CHAPTER FOUR

APPLICATION AND RESULTS

4.1 Scenario

CinreC is implemented as a web application. Guest visits the web pages of this web application. First of all, he/she registers to CinreC, determines a user name, password. He/she fills some profile information like occupation, birth date, gender, mail address, photo, real name if he/she wants. This data can be used to implement the content based filtering. Because men may like some specific films and women may like specific films. According to their ages, some films are chosen. Occupation is also connected with the films. Subjects may be connected with the occupations. After being registered, logins with user name and password. User starts to rank films with point between 0 and 5. User can also score with comma separated number. If a user gives score more than five or smaller than zero, the web page warns.

The web site randomly shows the films which the user has not ranked. The film name, the film photo and film release date are shown to users. If user wants, the active user can search with keywords. These keywords can be connected with user name, film name or actor name. The films are listed according to similarity between the item and keyword or at the left bottom of site some popular people of shortcuts are given. Users can also use these links. When an actor is searched, next page shows all films of this actor separately. Movies which this actor acted, movies which this actor became director or movies which this actor became scenarist are listed separately. User chooses a film from here and gives score. If a user wants, he can list movies by genre for finding the movies which he wants fast. The active user also ranks films from here. Users have to rank twenty films at least to get a recommend or to see their neighbors or nearest users to them.

There must be five mutual films between active user and his neighbor for affecting the prediction score. After user ranks twenty films at least, CinreC automatically calculates distance between five hundred users and active user. Active user can also increase this number by using 'new users' in find user page. In each

click, distances of seventy five new users are calculated. In the database, users are sorted according to ranked film count by descending. Seventy five new users are pulled from here in order. Generated users are sorted according to similarity distance by ascending. If active user wants, he/she can add to his/her friend list. He can follow their messages and the last films which they ranked. The active user can get recommendations anymore.

Recommendations are listed in four genres. The genres are chosen randomly. When the active user wants to get recommendations, genres are different from previous recommendations and items. These recommendations are computed with neighbors. These items are films which neighbors rank, the active user does not rank. This type recommendation is called user based collaborative filtering. In every film, there is a prediction score if CinreC has an enough data. A user can have a look at the film from CinreC before he has not seen the film. CinreC can give an opinion about the film. The other type is like that; when the active user is listed a film, the system shows her/him new films which are correlated to active film under “users who liked this item also liked”. This type recommendation is called item based collaborative filtering.

The active user can share his/her opinion in CinreC by writing a message. When a film is not found, anyone can add this film to the database via web page. Users can change their rankings whenever they want. Similarity distance is calculated again in this situation. Active user also can list the users according to the distance in ascending order. When it is clicked the any user, mutual films and the other films which are not seen can be seen and also the scores are examined, which score it was given, which score he gave. In this page comments of this user can be seen and this person can be added to the friend list or active user can go to another page for seeing the details of each film.

At the right side of the page, recommendations can be gotten according to the genre. Also users who rate the most are listed at the right side of the page. By clicking these users, each other can be examined. Number of movies and rankings

are shown at the right side. In each rating is given, similarity distance is calculated again. If another user different from active user gave score, when active user use 'new similarity button', similarity distance is updated between active user and that user.

In home page and recommendation page, recommendations are shown with prediction score and film information. Also last scores and opinions of the active user friends' exist at home page.

4.2 Data Set Description

In dataset which is experimented, there are 100,389 ratings, 7966 movies and 948 users. It is taken from grouplens.org. The data was collected during the seven-month period. Each user has ranked at least 20 movies. Ratings are between 0 and 5. Users' gender, age, occupation are in it. Movies' genre, release date, summary are also in it (Group Lens Research, 2006).

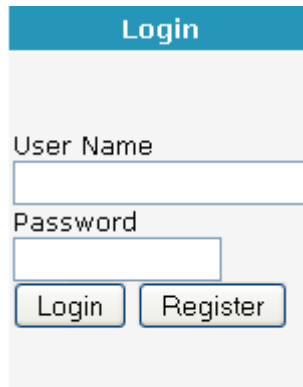
4.3 Used Technologies and Programming Languages

Microsoft Visual Studio Asp.Net 2008 with framework 3.5 web application, Microsoft SQL Server 2008, C-sharp, Ajax, Html, Css, JavaScript.

4.4 User Interface

CinreC is shown in a web page. It does not need to set up somewhere. There are some parts of this web page. Some parts are shown to registered users; some of them are shown to everyone. There are some special parts; these parts are only shown to system admin.

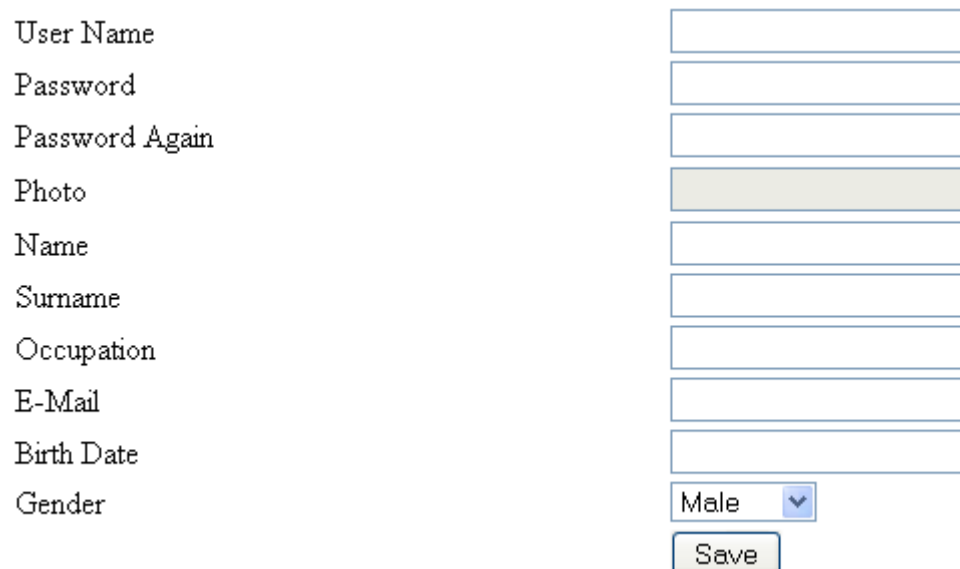
- User Login: It is shown to everybody for entering to CinreC. User can enter with username and password or register to CinreC by using 'register button'. It is also shown in Figure 4.1.



The image shows a login form with a blue header labeled 'Login'. Below the header, there are two text input fields: 'User Name' and 'Password'. At the bottom of the form, there are two buttons: 'Login' and 'Register'.

Figure 4.1 User Login

- **Register Page:** In this page, user can register to CinreC. First of all, he must register and use this page. For showing to other users, small photo, birth date, gender, occupation of the user are taken. It is thought that this information also gives to other users an idea about the taste. It is shown in Figure 4.2.



The image shows a registration form with the following fields and controls:


- User Name: Text input field
- Password: Text input field
- Password Again: Text input field
- Photo: File upload field (indicated by a grey box)
- Name: Text input field
- Surname: Text input field
- Occupation: Text input field
- E-Mail: Text input field
- Birth Date: Text input field
- Gender: Dropdown menu with 'Male' selected
- Save: Submit button

Figure 4.2 Register Page

- **Home Page:** When a user enters CinreC's url from browser, he sees this web page first. In the middle of it, movies which users added last are shown with name, imdb link, subject, small photo. At the right side, how many movies and rankings are there in the database are shown. Users who rank the most are shown at the right side. At the left side according to movie name, actor name, there are some searching tools. At the left

bottom side, there are some shortcuts for actors to go to their page. At the top middle side and at the bottom middle side, menu is shown. It contains user's manual for using web page easily. It is also seen in Figure 4.3.

If you want to watch a movie



User Guide | Home Page | Closest Users | Find Users | Give Score | My Movies | My Recommendations | Add New Movie

Login

User Name

Password

Login Register

Search Movie

Movie Name


Person

Search

mel gibson,julia roberts,
robert de niro,al pacino,

Movies that are added last


Blackadder Back & Forth



It's Millennium Eve and Blackadder is hosting a dinner party for a few select friends, Lady Elizabeth, Viscount George, Archbishop Melchett and Archbishop Darling. Baldrick devises yet another of his infamous cunning plans to help his ever greedy master Edmund Blackadder con money from his gullible friends. The pair build a 'time machine' from empty cereal packets and place bets with their friends as to when in history they will travel, retrieving various artifacts from their travels as proof, items which Blackadder already owns! However, in a strange twist of fate the time machine actually works and the pair are thrown back in history initially to the Jurassic period. Gradually the pair start to return to their own time stopping off at various famous times in history such as Sherwood Forest and the Battle of Waterloo but will they make it home

IMDb Link

Revolutionary Road



It's 1955. Frank and April Wheeler, in the seventh year of their marriage, have fallen into a life that appears to most as being perfect. They live in the Connecticut suburbs with two young children. Frank commutes to New York City where he works in an office job while April stays at home as a housewife. But they're not happy. April has forgone her dream of becoming an actress, and Frank hates his job - one where he places little effort - although he has never figured out what his passion in life is. One day, April suggests that they move to Paris - a city where Frank visited during the war and loved, but where April has never been - as a means to rejuvenate their life. April's plan: she would be the breadwinner, getting a lucrative secretarial job for one of the major

Wall

Total Movie Count:7966

Total Rate Count:100394

Users who are most active

nikaya - 183 film

nuray - 71 film


omer - 36 film

cetinozem - 33 film

Figure 4.3 Home Page

- **Movie Detail:** When active user wants to see the details of the movie, he sees this page. Name, director, scenarist, actors, imdb link, small photo, release date, genre, prediction score, subject of the movie, number of users who are used for calculating prediction score as in Figure 4.4 are shown. User can rank the movie from this page or update the ranking. If CinreC finds other movies which are liked by people who like this movie, it is shown at the bottom.

User Count:10
 Prediction Score:3,334 1964 Mary Poppins--Children--Comedy--Musical -



The movie combines a diverting story, songs, color and sequences of live action blended with the movements of animated figures. Mary Poppins is a kind of Super-nanny who flies in with her umbrella in response to the request of the Banks children and proceeds to put things right with the aid of her rather extraordinary magical powers before flying off again

[Imdb Link](#)

Director:robert stevenson

Scenarist:bill walsh,don dagradi

Cast:julie andrews,dick van dyke,david tomlinson,glynis johns

People who liked this movie also like these






	1992 - Aladdin-
	1965 - Sound of Music, The-
	1991 - Beauty and the Beast-
	1937 - Snow White and the Seven Dwarfs-
	1939 - Wizard of Oz, The-

Figure 4.4 Movie Detail

- **Recommendation Page:** In this page, if prediction score of a movie is more than 2.5, it is shown as in Figure 4.5. At most, four movies are shown.

Your Recommendations



	<p>Jean de Florette (1986)</p> <p>In a rural French village an old man and his only remaining relative cast their covetous eyes on an adjoining vacant property They need its spring water for growing their flowers, so are dismayed to hear the man who has inherited it is moving in. They block up the spring and watch as their new neighbour tries to keep his crops watered from wells far afield through the hot summer. Though they see his desperate efforts are breaking his health and his wife and daughter's hearts they think only of getting the water</p> <p>Imdb Link Puan:4,197</p>
	<p>Amateur (1994)</p> <p>Isabelle is an ex-nun waiting for her special mission from God. In the meantime, she is making a living writing pornography. She meets Thomas, a sweet, confused amnesiac who cannot remember that he used to be a vicious pornographer, responsible for turning his young wife, Sofia, into the world's most notorious porn queen. Sofia's on the run, convinced she's killed him. Together, Isabelle and Thomas set out to discover his past, a past waiting to catch up with him</p> <p>Imdb Link Puan:2,935</p>

Figure 4.5 Recommendation Page

- **Find User Page:** This page is made for calculating similarity measures between active user and other users. At first CinreC automatically calculates similarity distance of five hundred users. This web page is used

for finding more users whose the distance is known. To increase this number, for getting recommendation is very good. As shown in Figure 4.6 at the left side there are users whose distances are calculated. In that side, the first number shows order number, second one shows similarity distance and lastly is shown user name. Under each user, total movies and mutual movies are written. When active user clicks the ‘new similar button’, distances of seventy-five users are calculated. The closer to zero the better similarity distance is. At the bottom of button, there are some writings to tell what is done.

Users whose similarity distance are obtained

32 0,872 User:nuray 71 Mutual Movie Count: 44	<p>New Users</p> <p>You can determine distance between you and 75 users who are chosen individually by system. These users are added to database. This data is always updated after every action. It is better that similarity distance is close to zero</p> <p><i>Note: At the left side first number shows order number, the second one shows similarity distance.</i></p>
237 1,420 User: 185 Mutual Movie Count: 20	
250 1,442 User: 105 Mutual Movie Count: 12	
320 1,555 User: 269 Mutual Movie Count: 31	

Figure 4.6 Find User Page

- **Users Ordered by Similarity Page:** This page shows users who have a similarity distance with active user in order. There is some information about users. Number of total movies and mutual movies, gender and username of user are shown as in Figure 4.7.

Users Ordered by Similarity Distance

User	Order Number	Similarity Measure	Gender	Mutual Movie Count	Total Movie Count
	1	0,182	M	28	30
	2	0,471	M	7	51
	3	0,533	M	6	42
	4	0,540	F	5	33
	5	0,557	M	7	58
	6	0,600	M	11	155
	7	0,600	M	5	54
hayros	8	0,628	M	7	20

Figure 4.7 Users Ordered by Similarity Page

- **My Movies Page:** This page shows movies which active user has given a score. There is some basic information about the movies. Name, release date of the movie and ranking that active user did as in Figure 4.8. If that user wants to see more details about the movies, he can click that row and go to that web page.

Movie Name	Release Date	Score
Braveheart	1995	5
A moment to remember	2004	5
Godfather, The	1972	4,7
Professional, The	1994	4,5
Godfather: Part II, The	1974	4,5
Prestige	2006	4,5
Oldboy	2005	4,3
Good Will Hunting	1997	4,2
Spider-Man 2	2004	4,2
Black Swan	2010	4,1
Kill Bill: Vol. 2	2004	4,1
Scent of a Woman	1992	4,1
Inception	2010	4,1

Figure 4.8 My Movies Page

- **Film Rate Page:** In this page, movies with subject, name, release date, small photo and small place for giving score which active user has not given a score are listed randomly as shown in Figure 4.9. Active user can customize this page; movies can be listed by genre. If the user wants to see the details of the movie, by clicking the photo or name of movie he can see the details in the movie detail page.



Truckers form a mile long "convoy" in support of a trucker's vendetta with an abusive sheriff...Based on the country song of same title by C.W. McCall -

1978 - Convoy

After seven years in boarding school, Morgan Stewart is finally coming home. He discovers it's not the same happy home it used to be, so he decides to reform his social climbing, politically inclined parents and things get a little out of control-

1987 - Morgan Stewart's Coming Home

Kaydet ve Devam Et

Figure 4.9 Film Rate Page

- Detail of Other User Page: This web page shows the difference and similarity between active user and this user. It compares rankings and says which movies active user has not given a score. As shown in Figure 4.10, at the top side of page, similarity distance and order number of that user in point of distance are written. Under this, if active user wants, he can add this user to the friend list for following messages and last scores of this user. Under that, number of mutual movies is written and is shown these movies. At first, score of active user, after that score of that user and finally name of the movie with small photo are shown. Under this part, Number movies which active user has not given a score but this user has given a score are written. These movies are shown with name, small photo and score of this user. If active user wants to give a score these movies, he can click that row and go to movie details page. In this page also there are general comments of this user. Active user can follow these comments from his home page by adding this user to the friend list.

Order Number:2 Similarity Distance:0,471

Mutual Movie Count:7

You	Him	Movie
5	5	 Braveheart
4,5	4	 Professional, The
3,8	4	 Shawshank Redemption, The
4,7	4	 Godfather, The
3	3	 Clockwork Orange, A
4	5	 One Flew Over the Cuckoo's Nest
3,1	4	 Cool Hand Luke

Count of Movie that he has watched and you have not watched:44




Film	
 Dances with Wolves	5
 Jerry Maguire	5
 Being There	5

Figure 4.10 Detail of Other User Page

- **Famous Person Page:** In this page, at top, name of famous person is written. Under this, movies with name, release date and small photo which are connected with the actor are listed separately as shown in Figure 4.11. There are three categories; movies which this person became actor, movies which this person became scenarist, movies which this person became director. If active user wants to see details of any movie or give a score to any movie, he can click related row and go there. This page is prepared for reaching to the movies fast. Some people have watched movies, they don't remember the name of movie but they remember the actor, director or scenarist. It is very useful because of this.

Person : jack nicholson










Movies that he became director	Release Date
 Goin' South	1978
 Two Jakes, The	1990
 Goin' South	1978
Movies that he became scenarist	Release Date
 Head	1968
Movies that he became actor	Release Date
 Mars Attacks!	1996
 One Flew Over the Cuckoo's Nest	1975
 Batman	1989
 Wolf	1994
 Chinatown	1974

Figure 4.11 Famous Person Page

- **Add New Movie Page:** If the user can not find the movie that he wants to give score, he can add to database with this page. After approval of admin, this movie is listed to everyone. As shown in Figure 4.12, name, short summary, release date, imdb link, genres, photo of movie can be added to database. After this page, user can add directors, actors, scenarists of movie.


Movie Name	<input type="text"/>
Subject	<input type="text"/>
Release Date	<input type="text"/>
Imdb Link	<input type="text"/>
Genre	<input type="checkbox"/> Action <input type="checkbox"/> Adventure <input type="checkbox"/> Animation <input type="checkbox"/> Children <input type="checkbox"/> Comedy <input type="checkbox"/> Crime <input type="checkbox"/> Documentary <input type="checkbox"/> Drama <input type="checkbox"/> Fantasy <input type="checkbox"/> FilmNoir <input type="checkbox"/> Horror <input type="checkbox"/> Musical <input type="checkbox"/> Mystery <input type="checkbox"/> Romance <input type="checkbox"/> Thriller <input type="checkbox"/> War <input type="checkbox"/> Western <input type="checkbox"/> Bilinmeyen <input type="checkbox"/> SciFi <input type="checkbox"/> Family <input type="checkbox"/> Biography <input type="checkbox"/> Music
Photo	<input type="text"/> <input type="button" value="Gözet..."/>
<input type="button" value="Save"/>	

Figure 4.12 Add New Movie Page

- **Movie Detail Page without Login:** Users can want to see the details of a movie without login. Small photo, name, release date, imdb link, actor, director, scenarist, genres and subject of the movie are shown. Also users can see the other movies which people who like this film like. In top middle, in order there are release date, name, and genres of the movie. Under this part, at the left side, there are small photo and imdb link of the movie. At the right side, there is a summary of the movie. Under this part, director of the movie is shown. Under this, there is scenarist of the movie. Under this, there are actors of the movie. And as last, bottom of the page, there are movies that people who like this movie like. The aim of web site is to recommend some movies without login. For item based filtering, it is not necessary to give score to movies. We use the rates which exist in the

system. If somebody who liked a movie wants to watch another movie that he will like, he can use this page and he will get what he wants. Also in Figure 4.13, all these things which are told above are shown.

1995 Braveheart--Action--Drama--War --



[Imdb Link](#)


William Wallace is a Scottish rebel who leads an uprising against the cruel English ruler Edward the Longshanks, who wishes to inherit the crown of Scotland for himself. When he was a young boy, William Wallace's father and brother, along with many others, lost their lives trying to free Scotland. Once he loses another of his loved ones, William Wallace begins his long quest to make Scotland free once and for all, along with the assistance of Robert the Bruce

Director:mel gibson


Scenarist:randall wallace

Cast:mel gibson,sophie marceau,patrick mcgoohan,angus macfadyen


People who liked this movie also like these




1977 - Star Wars-




1993 - Fugitive, The-




1991 - Terminator 2: Judgment Day-



1980 - Empire Strikes Back, The-



1981 - Raiders of the Lost Ark-



1983 - Return of the Jedi-

Figure 4.13 Movie Detail Page without Login

- **My Friends Page:** This page shows the user's friend list. In other pages, user can add the other users to his friend list. Here they are shown. There is some basic information of them. Name, age of the user and how this user is closed to active user are seen. By clicking any row, user can see the details of that user. In that page, he can compare his movies with that user's movies. As seen in Figure 4.14, friends are ordered by similarity measure. Total film count of a friend, mutual film count of active user and friend can be seen in this page.

User	Order Number	Similarity Measure	Total Film Count	Mutual Film Count	Age
	6	0,600	155	11	40
hayros	8	0,628	20	7	21
	10	0,650	127	8	29
	17	0,733	44	9	29
	19	0,773	115	11	35
	21	0,800	124	8	38
	27	0,827	224	22	26
nuray	32	0,872	71	44	34
	36	0,890	133	20	19
	41	0,931	322	22	25

Figure 4.14 My Friends Page

- **Recommend a Movie According to Genre:** This part is used for getting recommendation according to genre. If the prediction score of a movie is bigger than three, it is shown to active user with name and small photo of the movie. As seen in Figure 4.15, new recommend is gotten by clicking the button 'Get Recommend'. After clicking the small photo or name of the movie, details of the movie can be seen in movie detail page.

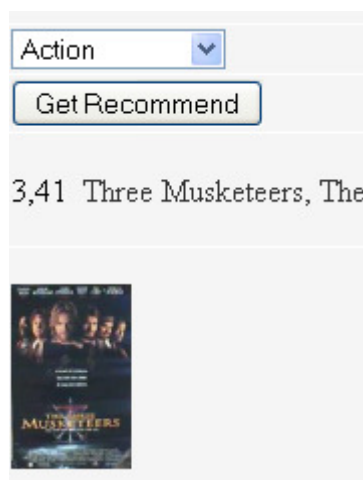


Figure 4.15 Recommend a Movie According to Genre

- **Comment Generally Part:** This part is for sharing opinions of people. These opinions can be about everything what people want to share. At bottom, last comments of active user's friends exist. They are ordered by date. People can follow their friends' last opinions from here as shown in Figure 4.16.

Your Comment:

Last Comments

22.05.2011 23:33:04 - nkaya
cinema paradiso is a grate film. All scenes are still in my mind. They are such as real.

10.05.2011 02:14:45 - nuray
"a moment to remember " izleyin harika bi film.

04.05.2011 14:39:49 - nkaya
kaybedenler klübü beklediğimin altında çıktı.

Figure 4.16 Comment Generally Part

- **Last Scores of Friends:** User can add friends to his friend list. As shown in Figure 4.16, he can follow their comments. As shown in Figure 4.17, he can follow his friends' last scores. In this part, user name, name and small photo of movie and score exist. If the active user wants to see the details of any movie, he can go there by clicking that row.









User	Movie	Score
nuray	 Cool Hand Luke	3,9
nuray	 Braveheart	5
nuray	 Cinema Paradiso	4,3
nuray	 Slumdog Millionaire	4,1
nuray	 Inception	4
nuray	 TRON: Legacy	3,1
hayros	 Memento	1
hayros	 Pretty Woman	4

Figure 4.17 Last Scores of Friends

- **How to Calculate Prediction Score Part:** This part explains to users the reason of prediction score. Scores of top ten closest users are used for this prediction. These ten users are ordered by similarity measure. As seen in Figure 4.18, order number, similarity measure, score, total film count, mutual film count, age of the user is shown.

User Count:10

User	Order Number	Similarity Measure	Score	Total Film Count	Mutual Film Count	Age
	6	0,600	4	155	11	40
	7	0,600	4	54	5	57
	9	0,640	4	37	5	22
	13	0,700	3	34	5	65
	14	0,711	3	159	9	44
	16	0,720	1	51	5	30
	16	0,720	4	239	5	40
	17	0,733	3	44	9	29
	18	0,771	4	29	7	23
	21	0,800	4	44	5	26

Tavsiye Puanı:3,338

 Kaydet

1997 Dante's Peak--Action--Thriller -



Volcanologist Harry Dalton and mayor Rachel Wando of Dante's Peak try to convince the city council and the other volcanologists that the volcano right above Dante's peak is indeed dangerous. People's safety is being set against economical interests

[Imdb Link](#)**Yönetmen:**roger donaldson**Senarist:**leslie bohem**Oyuncular:**pierce brosnan,linda hamilton,jamie renee smith,jeremy foley

Figure 4.18 How to Calculate Prediction Score Part

- **Give Score According to Genre Page:** It provides people to give score to the movies in definite genre. The movies are listed randomly. The release date, name and small photo of the movie are seen as shown in Figure 4.19. If active user wants to give score, he can click that row which is related and go to movie detail page.

Genre:Comedy








Release Date	Movie
2009	 The Hangover
2010	 New Kids Turbo
2004	 Seed of Chucky
1996	 High School High
1987	 Making Mr. Right
1997	 Love and Death on Long Island
1993	 Untamed Heart

Figure 4.19 Give Score According to Genre Page

4.5 Experimental Results

After the web site is published in the internet, internet users register to CinreC. They gave score to the movies. The prediction score that CinreC gives with real score which active user gives were compared. The movies which are given score by ten closest users at least were considered. The average deviation user by user is calculated. Some of them are shown in the following tables.

Total deviation is 33.11, total movie count is 52 and average deviation is 0.64.

This result is presented in Figure 4.20.

Prediction Score	Real Score	Difference	The Closest Similarity	The Farthest Similarity	Movie Name
3,236	2,4	0,836	0,533	0,905	Twelve Monkeys
3,155	3	0,155	0,182	0,8	Dead Man Walking
4,078	3,4	0,678	0,675	0,99	Usual Suspects, The
4,706	5	0,294	0,182	0,827	Braveheart
3,334	4,5	1,166	0,182	1,077	Professional, The
1,657	0,9	0,757	0,182	0,88	Pulp Fiction
4,143	3,8	0,343	0,182	0,872	Shawshank Redemption, The
3,144	3	0,144	0,182	0,96	Crow, The
3,399	3,5	0,101	0,182	1,244	Carlito's Way
2,404	2,1	0,304	0,6	0,95	Fargo
1,864	1,5	0,364	0,771	1,191	Striptease
4,194	4,7	0,506	0,182	0,773	Godfather, The
3,668	3,2	0,468	0,182	1,155	Reservoir Dogs
2,965	3	0,035	0,182	0,827	Top Gun
3,983	4	0,017	0,182	1,122	Cinema Paradiso
3,142	3	0,142	0,471	0,947	Clockwork Orange, A
3,029	1	2,029	0,182	1,071	GoodFellas
4,067	4,5	0,433	0,182	0,981	Godfather: Part II, The
3,229	2,8	0,429	0,711	0,947	Full Metal Jacket
3,526	3,5	0,026	0,182	0,827	Dead Poets Society
2,776	1,5	1,276	0,65	1,095	The Shining
3,98	4,2	0,22	0,182	0,866	Good Will Hunting
2,679	1,4	1,279	0,675	1,022	Heat
1,567	1	0,567	0,182	0,89	English Patient, The
3,327	2,9	0,427	0,182	1,012	Donnie Brasco
3,341	3,2	0,141	0,182	0,773	Liar Liar
3,349	3,1	0,249	0,182	0,872	Face/Off
3,348	3,5	0,152	0,64	0,89	Devil's Advocate, The
3,983	3,2	0,783	0,182	0,827	Titanic
4,031	1,2	2,831	0,823	1,096	Schindler's List
3,62	4	0,38	0,182	0,827	Conspiracy Theory
2,673	1	1,673	0,182	1,033	Kiss the Girls
2,783	2,3	0,483	0,182	0,827	Game, The
2,585	1,4	1,185	0,683	1,043	Boogie Nights
3,713	4	0,287	0,182	0,905	One Flew Over the Cuckoo's Nest
3,421	2,9	0,521	0,557	0,931	Ghost
3,042	2,8	0,242	0,182	1,073	Omen, The
2,488	1	1,488	0,182	1,096	Trainspotting
3,655	1	2,655	0,182	1,122	Lawrence of Arabia
3,502	3,1	0,402	0,471	0,969	Cool Hand Luke
3,609	4	0,391	0,182	0,969	Sleepers

Figure 4.20 Example one Comparison Real Score and Prediction Score

Total deviation is 7.946, total movie is 25 and average deviation is 0.318. This result is presented in Figure 4.21.

Prediction Score	Real Score	Difference	The Closest Similarity	The Farthest Similarity	Movie Name
4,222	4,3	0,078	0,3	0,5	Usual Suspects, The
4,914	5	0,086	0,12	0,46	Braveheart
4,615	4,6	0,015	0,171	0,517	Shawshank Redemption, The
4,704	5	0,296	0,12	0,54	Gone with the Wind
3,161	2,5	0,661	0,286	0,638	Basic Instinct
4,335	4,5	0,165	0,12	0,457	Top Gun
4,206	4,3	0,094	0,32	0,58	Cinema Paradiso
4,33	4,1	0,23	0,171	0,5	Groundhog Day
3,853	4	0,147	0,12	0,52	My Best Friend's Wedding
3,369	2,8	0,569	0,18	0,433	Men in Black
4,421	4,2	0,221	0,12	0,5	Good Will Hunting
3,958	4,1	0,142	0,12	0,375	Liar Liar
3,659	4,1	0,441	0,171	0,483	Face/Off
2,975	2,2	0,775	0,36	0,54	Devil's Advocate, The
4,64	5	0,36	0,12	0,36	Titanic
3,741	4	0,259	0,28	0,538	Conspiracy Theory
4,229	3,9	0,329	0,286	0,533	One Flew Over the Cuckoo's Nest
4,627	5	0,373	0,12	0,5	Ghost
3,763	4,1	0,337	0,171	0,4	Mission: Impossible
3,932	3,9	0,032	0,171	0,571	Cool Hand Luke
4,216	4,6	0,384	0,28	0,543	Primal Fear
4,018	4,3	0,282	0,44	0,578	Seven Years in Tibet
3,368	4	0,632	0,12	0,54	First Knight
4,137	5	0,863	0,12	0,52	Pretty Woman
4,822	5	0,178	0,18	0,433	Ransom

Figure 4.21 Example two Comparison Real Score and Prediction Score

Total deviation is 1.999, total movie count is 5 and average deviation is 0.4. This result is presented in Figure 4.22.

Prediction Score	Real Score	Difference	The Closest Similarity	The Farthest Similarity	Movie Name
4,577	4,5	0,077	0,2	0,7	Braveheart
3,914	5	1,087	0,2	1,1	Forrest Gump
4,176	4	0,176	0,5	0,9	One Flew Over the Cuckoo's Nest
3,811	4	0,189	0,5	1,3	Ran
3,529	4	0,471	0,5	1,1	Pretty Woman

Figure 4.22 Example three Comparison Real Score and Prediction Score

Total deviation is 8.62, total movie count is 10 and average deviation is 0.82. This result is presented in Figure 4.23.

Prediction Score	Real Score	Difference	The Closest Similarity	The Farthest Similarity	Movie Name
4,785	5	0,215	0,167	0,4	Braveheart
4,519	4	0,519	0,25	0,5	Shawshank Redemption, The
4,834	5	0,166	0,167	0,4	Forrest Gump
4,051	4	0,051	0,167	0,417	Jurassic Park
4,629	5	0,371	0,167	0,438	Terminator 2: Judgment Day
3,326	0	3,326	0,722	0,938	Godfather, The
3,124	0	3,124	0,857	1,167	Godfather: Part II, The
4,437	4	0,437	0,167	0,438	Terminator, The
4,128	4,5	0,372	0,25	0,583	The Shining
3,952	4	0,048	0,167	0,438	Jaws

Figure 4.23 Example four Comparison Real Score and Prediction Score

So far some examples were shown; real score and prediction score were compared. Average deviation was calculated. In Figure 4.24, average deviation according to each user is shown. Average deviation is always under 1. Generally it was observed when the movie count increases, average deviation decreases.

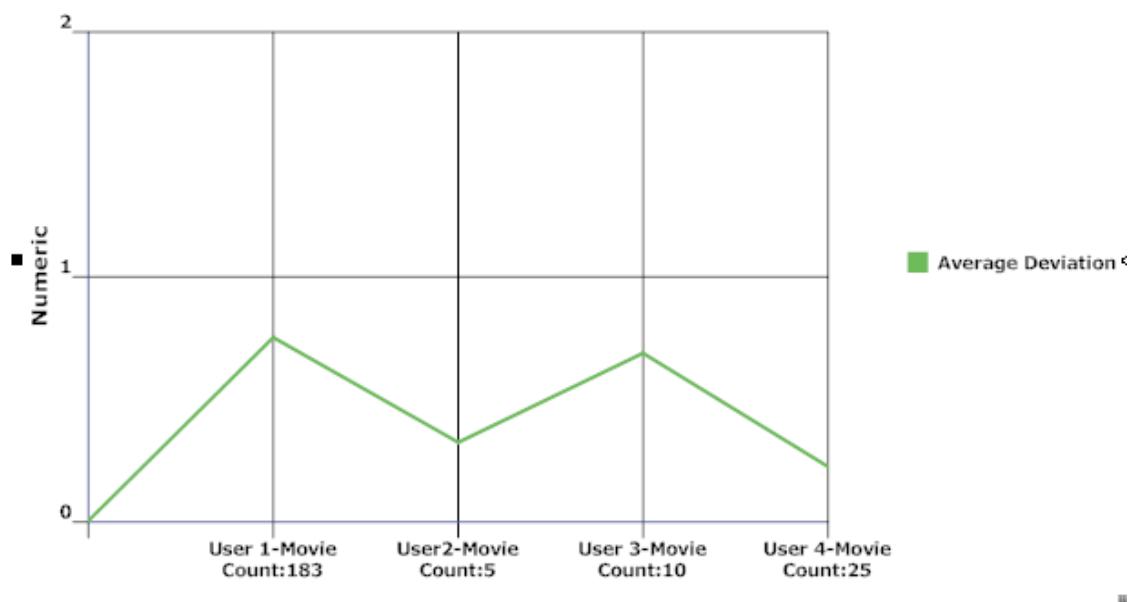


Figure 4.24 Average Deviation-Users

The amount of association rule with different support in implementing item based filtering are compared. This result was obtained as in Figure 4.25. When support increases, amount of association rule decreases.

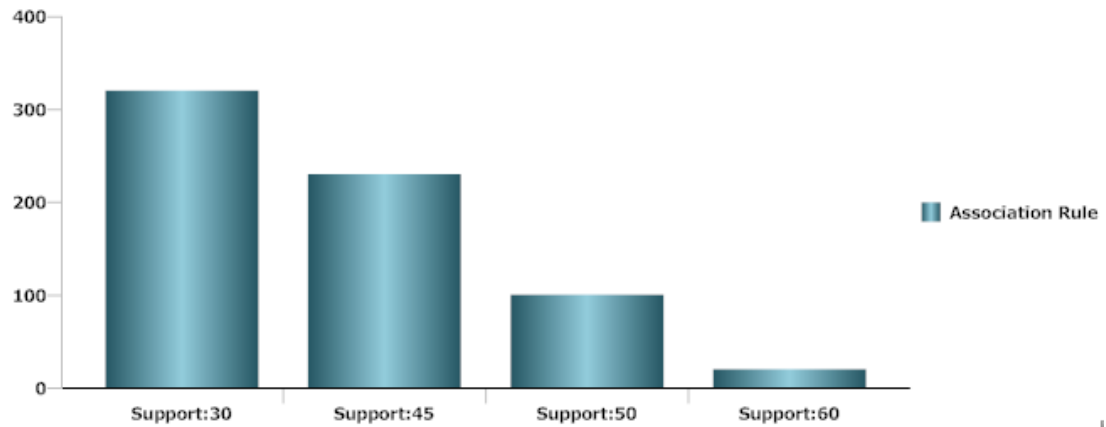


Figure 4.25 Support Amount-Association Rule

4.6 Comparison with Other Systems

CinreC was also compared with other recommendation systems. One of them that is used their dataset in this thesis is movielens.

- As shown in Table 4.1, CinreC is more equipped than MovieLens. It does not tell the reason of prediction score, just says the prediction score. It recommends movies according to similarity of actual movie.

	MovieLens	CinreC
User-Based CollaborativeFiltering	Yes	Yes
Item-Based CollaborativeFiltering	Yes	Yes
Show closer user list	No	Yes
Comment generally	No	Yes
Add user and follow	No	Yes
Recommendations by genre	No	Yes
Social Network	No	Yes

Table 4.1 MovieLens - CinreC comparison.

- Imdb : Imdb is the largest movie database in the internet. As seen in Table 4.2, Imdb uses item-based collaborative filtering, allow user to comment about movies.

	Imdb	CinreC
User-Based CF	No	Yes
Item-Based CF	Yes	Yes
Show closer user list	No	Yes
Comment generally	Yes	Yes
Add user and follow	No	Yes
Recommendations by genre	No	Yes
Social Network	No	Yes

Table 4.2 Imdb - CinreC comparison.

- Sinemalar.com: It is a web site that people can find details of new released movies. It is possible to find average score, summary, director, release date of movie and comment about movies by users. As seen in Table 4.3, except movie details, movie details and users' comments can be reached in sinemalar.com. And also there is a personal section for each user. Users can be added to friend list.

	Sinemalar.Com	CinreC
User-Based Collaborative Filtering	No	Yes
Item-Based Collaborative Filtering	No	Yes
Show closer user list	No	Yes
Comment generally	Yes	Yes
Add user and follow	Yes	Yes
Recommendations by genre	No	Yes
Social Network	Yes	Yes

Table 4.3 Sinemalar.Com- CinreC comparison

- Jester: It is an online joke recommender system. There are jokes as movies in CinreC. It does not use collaborative filtering, just uses principal

component analysis. There is no user list, no comment; they just list recommendations after given score to specific jokes. This method cannot be applied to movie recommender system. Because everybody must watch these specific movies. But this is really impossible. As seen in Table 4.4, jester is built for recommendation not a social network.

	Jester	CinreC
User-Based CF	No	Yes
Item-Based CF	No	Yes
Show closer user list	No	Yes
Comment generally	No	Yes
Add user and follow	No	Yes
Recommendations by genre	No	Yes
Social Network	No	Yes
Principal component analysis	Yes	No

Table 4.4 Jester - CinreC comparison

So far it was compared with recommender system and movie web site. Now it will be compared with some famous social networks.

- Facebook: It is widely used in the world. People add friend to their friend lists. They share everything about whatever they want. All friends in the friend list see this sharing. They can chat with each other, send messages and comment to sharing things. As seen in Table 4.5, as a recommendation, it uses content-based filtering. It follows the users where they click more than the others and these things appear more than the others.

	Facebook	CinreC
User-Based CF	No	Yes
Item-Based CF	No	Yes
Comment generally	Yes	Yes
Add user and follow	Yes	Yes
Recommendations	Yes	Yes
Social Network	Yes	Yes
Adding Module	Yes	No
Content-Based Filtering	Yes	No

Table 4.5 Facebook - CinreC comparison

- Twitter: In this system, people write some short messages and if somebody wants to follow him, he adds to follow list. According to short messages, it recommends some people to follow. As seen in Table 4.5, for doing this, it does content-based filtering. It examines the messages that user wrote.

	Twitter	CinreC
User-Based Collaborative Filtering	No	Yes
Item-Based Collaborative Filtering	No	Yes
Comment generally	Yes	Yes
Add user and follow	Yes	Yes
Recommendations	Yes	Yes
Social Network	Yes	Yes
Content-Based Filtering	Yes	No

Table 4.6 Twitter - CinreC comparison

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Some recommendations methods were described at the beginning of the thesis. Their disadvantages, advantages, challenges and how to overcome them were told. Some examples were given to be understood clearly. Collaborative filtering and association rule mining technique was combined (García et al., 2008) (Shi et al, 2008). Collaborative filtering was used for implementing user based filtering; association rule mining was used for implementing item based filtering. With the method that is used, user based filtering was provided to be fast. Experimental results also show, it is successful to calculate the prediction score. New equipped system among the recommendation systems was developed. Users can find closest users, comment about movies and get recommendations. All of them are together.

With CinreC, people will get good recommendations. The waste of time and the waste of money will be less than every time. They will know new people like thinking same about films. Good friendships will be made. If someone wants to watch a movie, but he/she does not decide it is good or bad idea to watch the movie, he/she will enter to this system. And CinreC will advise about the movie. It is like a guide.

Sometimes people want to laugh while they watch a movie, sometimes they want to cry, sometimes they want to be frightened, sometimes they want to be thrill. In CinreC they can get recommendation by genre.

CinreC has a system that is independent from items. It can be also applied on other item recommendation systems, such as book, song etc. It is very easy to convert to these systems.

5.2 Future Work

It was wanted to implement content based filtering with collaborative filtering (Lekakos & Caravelas, 2006) (Hannon et al., 2010) and to implement cross domain to say that for example, if you like this movie, you like this book (Winoto & Tang, 2008). But the time was not enough and the dataset was not stable for these. If these modules also are added to the project, recommendation system will be completed.

REFERENCES

- Abdelwahab, A., Sekiya, H., Matsuba, I., Horiuchi, Y., Kuroiwa, S., et al. (2009). Collaborative Filtering based on an iterative prediction method to alleviate the sparsity problem. *Information Integration and Web-based Applications & Services*, 250(10), 375-379
- Avancini, H., Candela, L., Straccia, U., et al. (2007). Recommenders in a personalized, collaborative digital library environment. *Journal of Intelligent Information Systems*, 1(1), 253-283.
- Billsus, D., & Pazzani, M. (1998). Learning Collaborative Information Filters, In Shavlik, J., ed., *Machine Learning: Proceedings of the Fifteenth International Conference*, Retrieved September 2010, 20, from <http://www.cs.wisc.edu/icml98/>.
- Cai, Z., Li, Z., Kang, Z., Liu, Y., et al. (2009). An Improved Algorithm of Apriori. *Computational Intelligence and Intelligent Systems*, 51(3), 427-428.
- Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y.S, Compton, P., Mahidadia, A., et al. (2010) Learning Collaborative Filtering and Its Application to People to People Recommendation in Social Networks. *AI 2010: Advances in Artificial Intelligence*, 6464(1), 476-485.
- Campos, L.M., Luna, J.M.F., Huete, J.F., Morales, M.A.R., et al. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning*, 51(7), 785-799.
- Chen, Z., Jiang, Y., Zhao, Y., et al. (2010). A Collaborative Filtering Recommendation Algorithm Based on User Interest Change and Trust Evaluation. *International Journal of Digital Content Technology and its Applications*, 14(9), 106-114.
- Gao, M., Wu, Z., Jiang, F., et al. (2011). UserRank for item based collaborative filtering recommendation. *Information Processing Letters*, 111(9), 440-446.

- García, E., Romero, C., Ventura S., Castro C., et al. (2008). An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *User Modeling and User-Adapted Interaction*, 19(1), 99-132.
- Goldberg, D., Nichols, D., Oki, B. M., Terry, D., et al. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM - Special issue on information filtering*, 35(12), 61-70.
- Goldberg, K., Roeder, T., Guptra, D.P.C, et al. (2001). Eigentaste: A constant-time collaborative filtering algorithm. *Information Retrieval*, 4(2), 133–151.
- Gong, S., Ye, H., Tan, H., et al. (2009). Combining Memory-Based and Model-Based Collaborative Filtering in Recommender System. *PACCS '09 Proceedings of the 2009 Pacific-Asia Conference on Circuits, Communications and System*, 30(11), 690-693.
- Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J., et al. (1999). Combining Collaborative Filtering With Personal Agents for Better Recommendations. In *Proceedings of the AAAI-'99 conference*, 22(10), 439-446.
- Grčar, M., Mladenič, D., Fortuna, B., Grobelnik, M., et al. (2006). Data Sparsity Issues in the Collaborative Filtering Framework, *Advances in Web Mining and Web Usage Analysis*, 15(9), 1-19.
- Group Lens Research(September 2006), <http://www.grouplens.org/node/73>.
- Guo, Q., Liu, J., Wang, B., et al. (2009). Improved collaborative filtering algorithm based on heat conduction, *Frontiers of Computer Science in China*, 3(3), 417-420.
- Guy, I., & Carmel D. (2011). Social recommender systems. *Recommender Systems*, 12(5), 283-285.

- Hannon, J., Bennett, M., Smyth, B., et al. (2010). Recommending twitter users to follow using content and collaborative filtering approaches. *Proceedings of the fourth ACM conference on Recommender systems*, 10(4), 99-206.
- http://en.wikipedia.org/wiki/Collaborative_filtering.
- Huang Z., & Zeng, D.D. (2011). Why Does Collaborative Filtering Work? Transaction-Based Recommendation Model Validation and Selection by Analyzing Bipartite Random Graphs. *Collaborative Filtering*, 4(1), 10-17.
- Kang, S., Choi, O., Baik, J., et al. (2010). Model-Based Dynamic Cost Estimation and Tracking Method for Agile software Development. *ICIS '10 Proceedings of the 2010 IEEE/ACIS 9th International Conference on Computer and Information Science*, 5(2), 743-748.
- Khabbaz, M., & LaksManan, L. (2011). Top Recs: Top-k Algorithms for Item Based Collaborative Filtering. *Proceedings of the 14th International Conference on Extending Database Technology*, 1(1), 213-224.
- Kumar, B.R.S, Ratnam, B.R., Babu, M.S.P, et al. (2010). Improvement of Personalized Recommendation Algorithm Based on Hybrid Collaborative Filtering. *International Journal of Computer Science & Communication*, 1(2), 429-432.
- Lecue, F. (2010). Combining Collaborative Filtering and Semantic Content-Based Approaches to Recommend Web Services. *ICSC '10 Proceedings of the 2010 IEEE Fourth International Conference on Semantic Computing*, 3(1), 200-205.
- Lekakos, G. & Caravelas, P. (2006). A hybrid approach for movie recommendation. *Multimedia Tools and Applications*, 36(1), 55-70.
- Liu, N.N., Zhao, M., Xiang, E., Yang, Q., et al. (2010). Online evolutionary collaborative filtering. *Proceedings of the fourth ACM conference on Recommender systems*, 2(1), 95-102.

- Liu, Z., Qu, W., Li, H., Xie, C., et al. (2010). A hybrid collaborative filtering recommendation mechanism for P2P networks. *Future Generation Computer Systems*, 26(8), 1409-1417.
- Martinez, A.B.B., Montenegro, E.C., Burguillo, J.C., Lopez, M.R., Fonte, F.A.M, Peleteiro, A., et al. (2010). A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Information Sciences: an International Journal*, 180(22), 4290-4311.
- Mikoaj, Morzy, T. , Królikowski, Z., et al. (2004). Incremental Association Rule Mining Using Materialized Data Mining Views. *Advances in Information Systems*, 3261(2005), 77-87.
- Para-Santander, D., & Brusilovsky P. (2010). Improving Collaborative Filtering in Social Tagging Systems for the Recommendation of Scientific Articles. *Web Intelligence and Intelligent Agent Technology*, 1(2010), 136-142.
- Redpath, J., Glass, D.H., McClean, S., Chen, L., et al. (2010). User Based Collaborative Filtering: Sparsity and Performance. *Proceeding of the 2010 conference on STAIRS 2010: Proceedings of the Fifth Starting AI Researchers' Symposium*, 2(1), 264-276.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *CSCW '94 Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1(1), 175-186.
- Rohini, U., & Ambati V. (2006). Improving Re-ranking of Search Results Using Collaborative Filtering. *Information Retrieval Technology*, 4812(2006), 205-216.
- Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J., et al. (2000). Application of Dimensionality Reduction in Recommender System-A Case Study. In *ACM WebKDD 2000 Workshop*.

- Sarwar, B., M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J., et al. (1998). Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. *CSCW '98 Proceedings of the 1998 ACM conference on Computer supported cooperative work*, 27(2), 345-354.
- Shardanand, U., & Maes, P. (1995). Social Information Filtering: Algorithms for Automating 'Word of Mouth'. *CHI '95 Proceedings of the SIGCHI conference on Human factors in computing systems*, 1(1), 210-217.
- Shi, X., Ye, H., Gong, S., et al. (2008). A Personalized Recommender Integrating Item-Based and User-Based Collaborative Filtering. *ISBIM '08 Proceedings of the 2008 International Seminar on Business and Information Management*, 1(1), 264-267.
- Sun, J., Zhao, J., Yu, X., et al. (2009). A Collaborative Filtering Algorithm with Phased Forecast, Rough Sets and Knowledge Technology. *RSKT '09 Proceedings of the 4th International Conference on Rough Sets and Knowledge Technology*, 1(1), 698-705.
- Winoto, P. & Tang, T. (2008). If you like Devil Wears Prada the Book, Will You also enjoy the Devil Wears Prada the Movie? A Study of Cross-Domain Recommendations. *New Generation Computing*, 1(1), 209-225.