THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

CONTENT BASED MOVIE RECOMMENDATION USING TURKISH MOVIE DESCRIPTIONS

Master's Thesis

ELİF GÜNER

ISTANBUL, 2019

REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

COMPUTER ENGINEERING

CONTENT BASED MOVIE RECOMMENDATION USING TURKISH MOVIE DESCRIPTIONS

Master's Thesis

ELİF GÜNER

Supervisor: ASSIST. PROF. DR. TEVFİK AYTEKİN

ISTANBUL, 2019

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

Name of the thesis: Content Based Movie Recommendation Using Turkish Movie Descriptions
Name/Last Name of the Student: Elif Güner
Date of the Defense of Thesis: 29 May 2019

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assist.Prof. Dr. Yücel Batu SALMAN
Graduate School Director

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Sciences.

Assist. Prof. Dr. Tarkan AYDIN
Program Coordinator

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Sciences.

Examining Comittee Members	Signature
Thesis Supervisor Assist. Prof. Dr. Tevfik AYTEKİN	----------
Member Prof. Dr. Alper TUNGA	----------
Member Prof. Dr. Yücel SAYGIN	----------

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Tevfik Aytekin for his support and guidance throughout my thesis.

I would also like to thank my family for all of their help and support.

Finally, I would like to thank to my husband, Hüseyin Güner, for his eternal love and endless support.

İstanbul, 2019 Elif GÜNER

ÖZET

TÜRKÇE AÇIKLAMALAR KULLANAN
İÇERİK TABANLI FİLM ÖNERİ SİSTEMİ

Elif GÜNER

Bilgisayar Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi Tevfik AYTEKİN

Mayıs 2019, 45 sayfa

Öneri sistemleri kullanıcılara ilgilerini çekebilecek ürünleri sunmayı sağlayan filtreleme teknikleridir. Günümüzde çok büyük miktarlardaki veri içinden kullanıcıların tercih yapması oldukça zordur. Bu durumda kullanıcılara doğru ürünleri sunmayı sağlayan öneri sistemleri günümüzde daha da önem kazanmıştır.

Bu çalışma çeşitli film öneri sistemlerinin başarısını ölçmeyi hedeflemektedir. Öneri sistemlerinin başarısı, kullanıcıların filmlere verebilecekleri puanlara en yakın puanları tahmin edebilmeleriyle ölçülür. Bu çalışmada kullanılan öneri sistemlerinden biri olan içerik tabanlı filtreleme için bir Türkçe sinema web sitesinden veri alınmış, Türkçe kelimeler işleme sokularak metodun performansı içerik tabanlı filtreleme sisteminin türleri olan özet tabanlı filtreleme ve üstveri tabanlı filtreleme için ölçülmüştür.

Aynı veri seti ile çeşitli işbirlikçi filtreleme sistemlerinin de performansı ölçülmüş, içerik tabanlı filtreleme sistemleri ile doğru tahmin yapabilme performansları karşılaştırılmıştır.

Anahtar Kelimeler: Öneri Sistemleri, İçerik Tabanlı Filtreleme, İşbirlikçi Filtreleme

ABSTRACT

CONTENT BASED MOVIE RECOMMENDATION
USING TURKISH MOVIE DESCRIPTIONS

Elif GÜNER

Computer Engineering

Thesis Supervisor: Asst. Prof. Dr. Tevfik AYTEKİN

May 2019, 45 pages

Recommendation systems are filtering techniques that enable users to present products that may be of interest to users. Nowadays it is very difficult for users to choose from a large amount of data. In this case, suggestion systems that provide the right products to the users have become more important today.

This study aims to measure the success of various film recommendation systems. The success of recommendation systems is measured by users' ability to estimate the closest ratings they can give to the films. For the content-based filtering, which is one of the recommendation systems used in this study, data was retrieved from a Turkish cinema website and Turkish words are preprocessed. The performance of the plot based filtering and metadata based filtering methods which are the types of content based filtering methods were measured.

With the same data set, the performances of various collaborative-based filtering systems were also measured, and accuracy of content-based filtering systems and collaborative-based filtering systems estimation performances were compared.

Keywords: Recommendation Systems, Content Based Filtering, Collaborative Based Filtering

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

API	:	Application Programming Interface
BMF	:	Biased Matrix Factorization
IDE	:	Integrated Development Environment
IDF	:	Inverse Document Frequency
MAE	:	Mean Absolute Error
NLP	:	Natural Language Processing
ORM	:	Object Relational Mapping
RC	:	Remote Control
RMSE	:	Root Mean Square Error
SC-AFM	:	Sigmoid Combined Asymmetric Factor Model
SI-AFM	:	Sigmoid Item Asymmetric Factor Model
SSVD++	:	Sigmoid SVD Plus Plus
SU-AFM	:	Sigmoid User Asymmetric Factor Model
SVD	:	Singular Value Decomposition
SVD++	:	SVD Plus Plus
TF	:	Term Frequency

1. INTRODUCTION

Recommendation systems aim to provide appropriate and personalized suggestions of products to users. In the big data era, it is hard for users to decide which product to buy, which music to listen or which movie to watch between large amounts of data. Recommendation systems help users to easily choose the products of their interest.

Two types of movie recommendation systems are investigated in this study. These are content based and collaborative filtering models. Content based filtering methods for movie recommendation has two types. One is based on description of the movies and the other one is based on actors, director, genre and keywords information of the movies. These models find the similarities between the movies using description or metadata information. Based on the rating information of the users, rating prediction is done using the most similar movies to the target movie. There are many types of collaborative filtering models. Most important types are user based and item based collaborative filtering methods. In these methods, similarities are calculated based on rating information.

During the literature research, it is observed that there are many studies for the content based movie recommendation systems for the other languages, but there are very limited studies for the Turkish language. The aim of this study is to investigate the rating prediction performance of a Turkish movie web site based on the content based filtering model using plot descriptions in Turkish language. Prediction success of metadata and keyword based method which is another type of content based filtering model is also measured. In order to measure the prediction performance of the collaborative filtering models, MyMediaLite open source framework is used with the same dataset that is received from the Turkish movie web site.

In the second section of this thesis, general information is given about content based filtering and collaborative filtering recommendation system models. Also, other studies about movie recommendation systems are described. In the third section, creation of the

dataset, collection of the data and methodologies that are followed while implementing the project which is prepared for measuring the accuracy success of the content based filtering is explained. Results of the experiments that are performed for content based and collaborative filtering methods are displayed in tables and graphics in the fourth section. Results are evaluated in the last section.

2. THEORETICAL FOUNDATIONS AND RESOURCE RESEARCH

2.1 RECOMMENDER SYSTEMS

The rapid growth of the Internet has led to a new era of information. It has a big impact both on the academic research and on the daily life. There was a revolution of the way information is collected, stored, processed, presented, shared and used. There are plenty of data files in the form of text, picture and video and they are easily accessible. However, easily accessible does not mean that easily found. Users need to cope with situations where they have too many options. The tremendous quantity of information that exists on the Internet is hard to be classified and used by a simple user. As a result, people need assistance in order to limit their preferences effectively and quickly from the infinite available possibilities. So, recommender systems have been developed in order to propose web pages, restaurants, books, movies and so on. In order to provide good recommendations, recommender systems are based on modeling the content, the social groups the user belongs to and the end user preferences (Christakou & Stafylopatis, 2005).

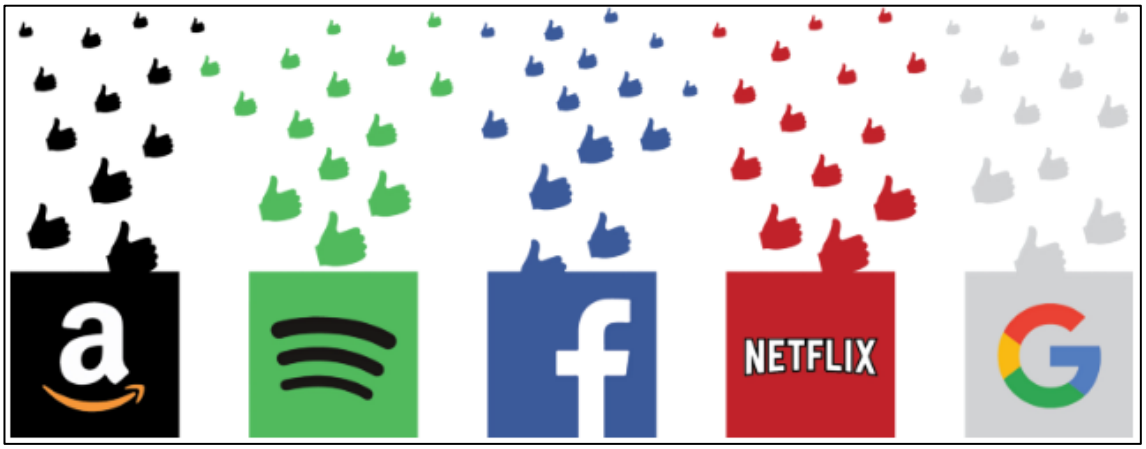
Recommendation systems are software and techniques that provide suggestions for a user. Suggestions are produced using a number of techniques, such as which products can be sold, which music can be listened, which books to read and which films to watch. Recommendation systems are primarily used to prevent inexperienced people from making comparison between thousands of alternative products. Amazon.com, which is popular around the world, uses recommendation systems to personalize the online shop for each customer. As the recommendations are personalized, each user or user group receives different suggestions. There are also non-personalized suggestions which is much simpler to produce them. Examples are the lists of the top ten products elected by the editors. Non-personalized suggestion systems are useful in some cases but are not considered under recommendation systems. Basically, personalized recommendations are presented as ordered lists. When this is done, the recommendation systems try to estimate the most appropriate products based on the user's preferences and choices. In order to make such a proposal, systems collect information about the user either by rating the

products such as scoring or by interpreting the user's movements. For example, the recommendation system can interpret the user's movements and direct the user to other products (Durukan, 2018).

The development of recommendation systems has started with simple observations. Individuals generally rely on the advice of others when making routine and daily decisions. For example, when deciding which movie to watch, we rely on the criticism of the movie and the comments of the other people, the letters of recommendation when hiring, the advice of the friend you trust when choosing books and we usually apply these recommendations. Suggestion systems use techniques and algorithms to propose suggestions from the user community to a relevant user by trying to emulate such behaviors. Recommendations are for the products that similar users like. According to this method, known as behavioral filtering, if the active user previously liked some common products with some of the other users, other recommendations and tastes from these users should be interesting for the active user. Recommendation systems have proved to be a valuable tool to overcome the overloading problem. After all, the suggestion systems can direct the user to new and previously uninformed products according to their interests. Recommendations are produced based on user data that is stored in the database, appropriate products and user's previous movements according to the user's needs and choices. The user can look at these suggestions and make decisions. User may like or dislike the suggestions, can use it immediately or in the next stage, but all actions and transactions of the user are saved in the database for later use (Durukan, 2018).

Recommendation systems are quite new compared to other research areas. In the mid-1990s, it emerged as an independent research area. It has an important role in the world-wide web sites such as YouTube, Facebook, Amazon and Netflix as in Figure 2.1. Many media companies offer recommendation systems as a service to their subscribers. For example, Netflix, an online movie rental service, gave a million-dollar prize to the first team that improves the success of their recommendation system by 10 percent.

Figure 2.1: Web sites that are actively using recommendation systems



Source: Durukan, 2018

Recommendation systems are usually the systems used by the companies which make trade and sales to increase profits. Attention is paid to users by carefully selecting the products and thus increasing sales volume and profits. General techniques for this purpose are similarity, novelty, serendipity and diversity.

The most obvious purpose of recommendation systems is to give the user the most personalized recommendations. Because users are more likely to buy and consume the products they find interesting. Similarity is not sufficient alone, even if it is the most obvious goal.

Diversity refers to different types of items in a recommendation list. Recommendation systems often suggest products that are most liked by other users. However, the similarity of the recommended products may cause users to be overwhelmed by these suggestions. If the recommendations contain different types of products, these suggestions are more likely to be liked by the users. This type difference sometimes determined by the content (e.g. genre information for movies). Diversity is an important feature to ensure that the user is not always recommended the same products.

Novelty refers to recommending unknown items to the user. Recommendations are more effective if the user has not seen the recommended products before. For instance, the popular films which are recommended usually are not found interesting by the users.

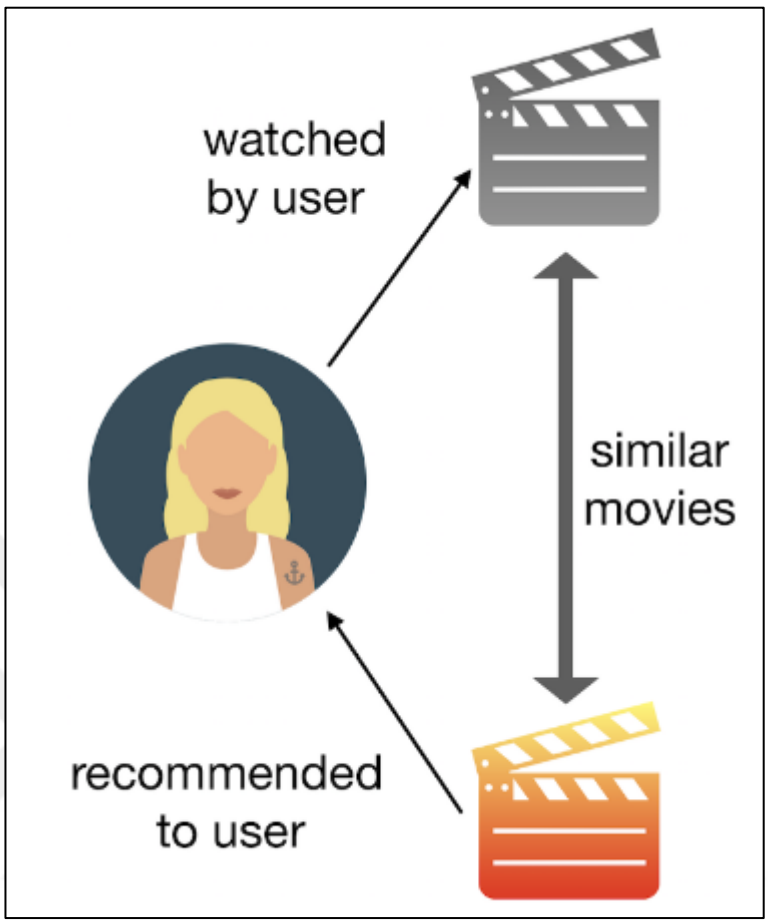
Serendipity refers to recommending unexpected and useful items to the users. The unexpected and intriguing product attracts the attention of the user and makes the user to feel like discovering. The serendipity feature is often to suggest a product that is truly surprise the user, rather than a product that the user has not previously known. For instance, if a new Italian restaurant is opened in the neighborhood, it is a novelty to recommend this restaurant to someone who eats Italian food, but it is not serendipity. On the other hand, if this restaurant is recommended to a Chinese person and that person does not expect this kind of suggestion, than it is serendipity. Serendipity does not only increase sales, but also launches a new stream. Due to the discovery of completely new interests, it provides long-term benefits for the seller.

Recommendation systems are mainly based on two methods called content based filtering and collaborative filtering. Also, there are studies about hybrid recommendation systems which combines these two methods.

2.1.1 Content Based Filtering

In the content based filtering methods, the similarity between two items is determined using the content of the items. The content can be any available data about the item. If the item is a movie, than the content may be plot description or cast, director, genre and keywords. The general idea is, if an individual liked a particular item in the past, he will like an item that is similar to it as well as shown in Figure 2.2.

Figure 2.2: Content based filtering model



Source: Durukan, 2018

The main advantage of content based filtering method is new item problem. When a new item comes into the system, its similarity with the other movies can be calculated using its content. Than, most similar movies would be recommended which the users gave high ratings. On the other hand, there would be less diversity in the recommendations since the item similarity is the only criteria.

In order to measure the prediction success of the plot description content based filtering method, summary or description of the movies are used. First, description information of the movies are preprocessed and roots of each word is found. Then, document vectors are created for each movie. At this point, term frequency of each word is calculated using TF-IDF (Term Frequency-Inverse Document Frequency).

The relative frequency of a word in a document is called TF and it is calculated as the ratio of count of a term in a document to count of all the words in the same document. The IDF value specifies that the frequently used terms do not contain information and it acts as a filter. In the following formula, *N* stands for total number of documents, *dft* stands for the number of documents that includes term *t, idft* shows how much information term *t* includes :

$$idf_t = log_{10} \frac{N}{df_t} \tag{2.1}$$

By combining the terms, the weight of each term is determined for each document as TF-IDF:

$$(tf - idf)_{t,d} = tf_{t,d} * idf_t \tag{2.2}$$

Count method is another type of content based filtering method. In this method, frequency of a word in a document is calculated instead of TF-IDF. This method generally gives worse results comparing to TF-IDF since TF-IDF down weights the words that don't make much sense.

Another type of content based filtering method is metatada based. Actors, director and genre as well as keywords that are extracted from the summary of the movie by TF-IDF method are used as content in this method. All these data is put to corpus and document vectors are created. If the feature *j* exists in the movie *i* content, than movie[*i*] feature[*j*] is 1, else it is 0. Document vectors are created this way and similarities between the movies are calculated using these vectors.

Several methods can be used to calculate similarity. Some of them are the euclidean, the Pearson and the cosine similarity measures. There is not an exact answer to which score

would be the best. Different methods may work better in different scenarios. In this thesis, cosine similarity is used to calculate the numbers that defines the similarity between two movies. It is also preferred for being relatively easy and fast. The formula for cosine similarity is defined as follows:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.3}$$

After all the similarity scores between the movies are calculated, similarity matrix is created. Rating predictions are done considering the highest n correlations in the matrix. Example of a similarity matrix can be seen in Figure 2.3.

Figure 2.3: Similarity matrix

	$Movie_1$	$Movie_2$	$Movie_3$	$\cdots$	$Movie_n$
$Movie_1$	1	0.158	0.138	...	0.056
$Movie_2$	0.158	1	0.367	...	0.056
$Movie_3$	0.138	0.367	1	...	0.049
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$Movie_n$	0.056	0.056	0.049	...	1

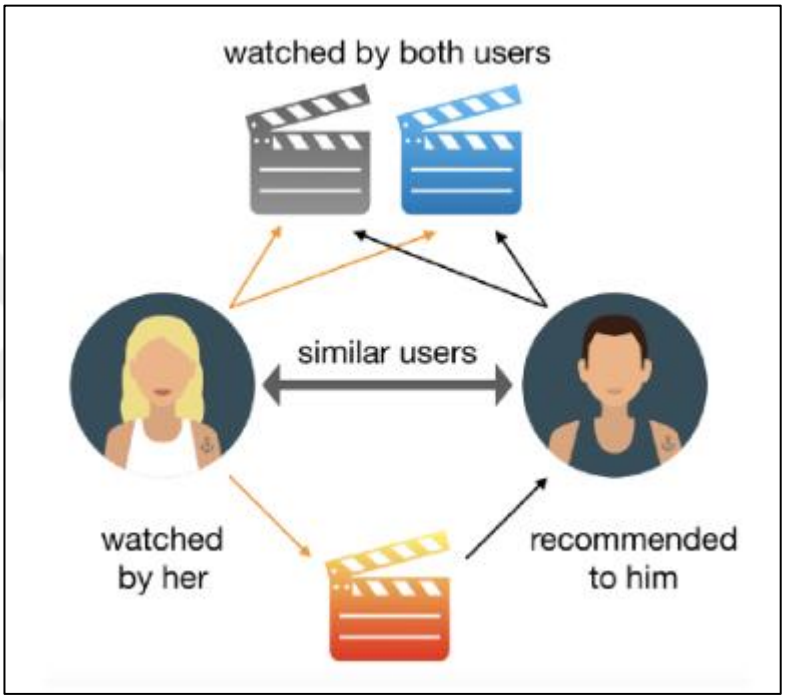
Similarity matrix.

2.1.2 Collaborative Filtering

Collaborative filtering model is one of the most widely used recommendation method. People with similar interests are matched and recommendations are provided based on this matching. Recommendations are usually extracted from the correlations of elements that are explicitly estimated by different users or statistical analysis of the patterns. The similarity between users is calculated instead of the elements. Rating of the unknown elements are predicted by using a combination of the nearest neighbors' results. Even the user provides only a few ratings, recommendation quality is generally high (Christakou & Stafylopatis, 2005).

Collaborative filtering model is based on users' ratings. It recommends the movies that users haven't watched yet, but similar users have already watched and liked. This model takes into account the movies both of them watched and rating scores of them in order to decide if two users are similar or not. It predicts the rating of a movie using the similar users' ratings, for a user who hasn't watched that movie yet.

Figure 2.4: Collaborative based filtering model



Source: Durukan, 2018

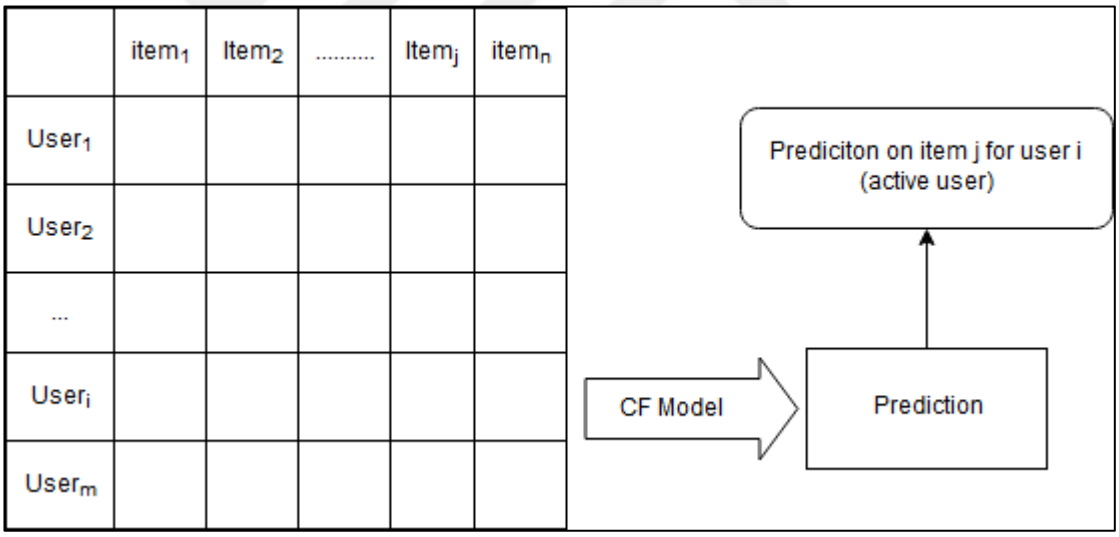
In order to get successful results, ratings are necessary for collaborative filtering model. But, all users do not always rate the items. Moreover, some of them never rate any of them. This can lead to cold-start problem. But, diversity in the suggestions can be considered as an advantage of this method. For example, user X may like comedy and romantic movies. User Y likes comedy movies but never watched romantic movies. This method will recommend romantic movies to user Y, based on the common taste that the two users have for comedy movies. If user Y figures out that he likes romantic movies, he can add a lot of new movies to watch on his list. In this case, diversity of this model

makes the recommendation successful. Or, If user Y figures out that he does not like romantic movies, it means that recommendation is not successful (Grimaldi, 2018).

2.1.2.1 Memory based

Memory based methods are also known as neighborhood-based collaborative filtering systems. These are among the oldest recommended systems of user-item rating that are known to be neighbors. The neighborhood relationship can be explained in two ways, user based and item based. Memory based methods do not work well if the user-item rating matrix is too sparse. In this method, entire database is loaded into system memory and predictions are made based on such in-line memory database. Huge data is a problem of this method (Do, Nguyen & Nguyen, 2010).

Figure 2.5: User item rating matrix



2.1.2.1.1 *Item based*

In this filtering technique, predictions are made based on the similarity between items. All items rated by an active user are received from the user-item matrix. Then, similarity between the received items and the target item is calculated and k most similar items are selected. Then, weighted average of the active users' rating is taken on the most similar k items.

Since the similarity between items is used, item based collaborative filtering looks like content-based filtering model. Item based model generally provides better predictions than the user based model. It is also more stable than the user based model. So, it allows pre-calculation and improves online performance (Ricci, 2015).

2.1.2.1.2 *User based*

User based collaborative filtering technique is based on the idea that similar users would have similar taste. If user A and user B have watched the same movies and they rated them nearly the same. But, user B has not watched movie X yet, but user A did. If user A likes that movie, it can be assumed that user B will like it as well. User based collaborative filtering is based on this model and recommends items by finding similar users to the active user whom a movie will be recommended (Schafer, Frankowski, Herlocker & Sen, 2007).

This model is more dynamic than the item based model. So, pre-calculation may cause poor predictions since the similarity between users may differ even a few ratings change (Ricci, 2015). Another disadvantage of this model is cold start problem which new users will not have enough ratings in order to be compared with other users.

2.1.2.2 Model based

In model based method, huge database is compressed into a model and predictions are made by applying reference mechanism into this model. Model based collaborative filtering can response user's request immediately (Do et al., 2010).

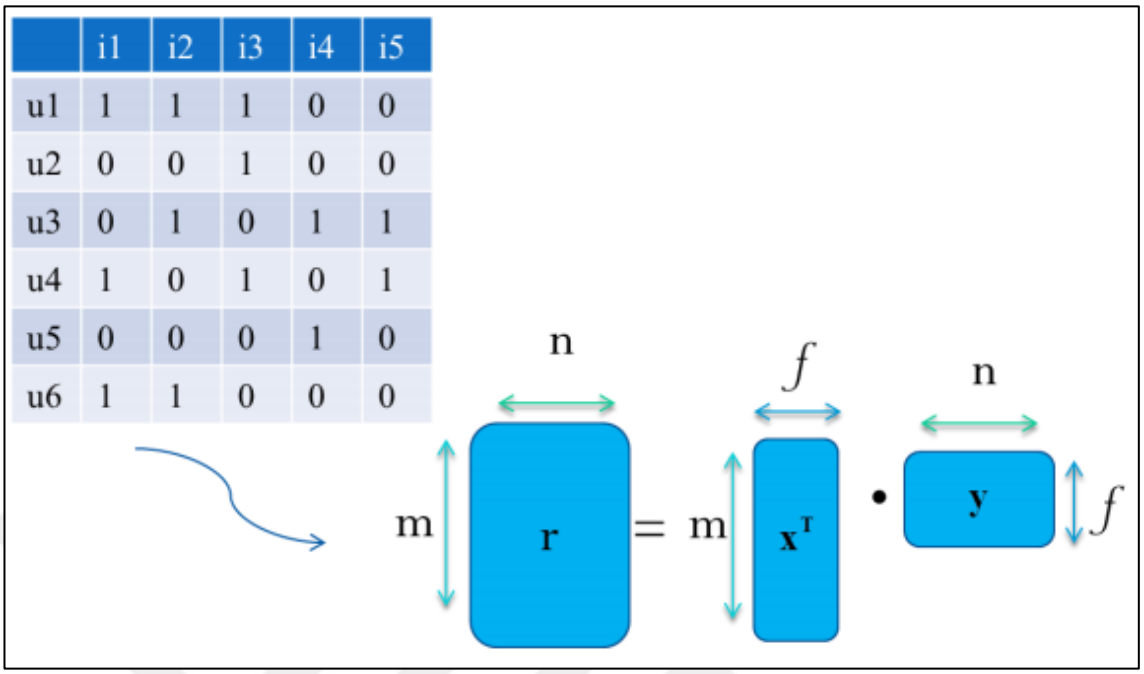
2.1.2.2.1 Singular Value Decomposition (SVD)

SVD is a matrix factorization model which is used for constructing a feature matrix for users and for items. Matrix factorization model which is one of the leading types of latent factor models, can make more successful predictions comparing to the nearest neighbor methods. Neighborhood methods focus on the similarity between items or users, so they are able to discover localized relationships (e.g., someone who likes Xman also likes Ironman), but they are not good at finding out the user's overall taste. Latent factor models try to define the ratings by items and users and make them directly comparable by transforming both items and users to the same latent space. This model is able to predict the comprehensive structure (e.g., a user likes thriller movies) but is not good enough in analyzing associations among small sets of closely related items (Guan, Li & Guan, 2016).

Latent factor model also handles the scalability and sparsity problems which exists in neighborhood based collaborative filtering. In this model, recommendation problem is turned into an optimization problem (Huang, 2018) .

Latent factor is a concept which describes a feature that a user or an item have. For instance, in case of music, latent factor may refer to the genre of that music. SVD reduces the dimension of the user-item matrix by extracting its latent factors. Each user and each item is mapped into a latent space with dimension r. So, the relationship between users and items would be more recognizable as they become directly comparable (Huang, 2018).

Figure 2.6: Decomposition of rating matrix into user and item factor matrices



The matrix factorization can be thought of as finding two matrices that are the product of the original user-item matrix. Vector `qi` represents each item and vector `pu` represents each user. The dot product of these vectors represents the expected rating.

$$expected\ rating = \hat{r}_{ui} = q_i^T p_u \tag{2.4}$$

`qi` and `pu` vectors can be detected by making the square error difference between the expected rating and the actual rating in the user-item matrix minimum.

$$minimum(p,q) \sum_{(u,i)\epsilon K} (r_{ui} - q_i^T . p_u)^2 \tag{2.5}$$

In order to prevent overfitting problem and increasing the generalization performance, a regularization factor λ is added to above minimization equation.

$$minimum(p,q)\sum_{(u,i)\epsilon K}(r_{ui}-q_i^T.p_u)^2+\lambda(\|q_i\|^2+\|p_u\|^2) \quad (2.6)$$

If a user gives a low rating to a movie and if there is no other rating from this user, this algorithm would give `qi` a large value for minimizing the error. This would cause all ratings from this user to other movies to be very low. This is an unsolicited status. By adding the regularization factor to the equation, giving vectors large value will minimize the equation and this kind of situations will be prevented (Golub & Reinsch, 1971).

2.1.3 Hybrid Filtering

Hybrid filtering methods combine several recommendation techniques in different ways like, merging content based filtering with collaborative filtering. Both collaborative filtering and content based methods have their own advantages and disadvantages. Hence when both of them is combined together, then the advantage of both techniques can be used to overwhelm the disadvantages of others. For instance, if a user has no rating, collaborative filtering method can not make prediction since it uses rating matrix. In this case, content based filtering can be used for recommendations (Burke, 2007).

Hybrid filtering methods are implemented in multiple ways like by making collaborative based and content based recommendations separately and then combining them, by adding collaborative based abilities to content based techniques and vice versa or by merging the techiques into one model. Most of the studies use hybrid filtering system in a weighted way (Çano & Morisio, 2017). According to the studies, hybrid filtering methods are more successful comparing to the pure content based or pure collaborative based models.

Netflix is an example of the hybrid filtering methods. It compares the watching and searching routines of similar users (i.e., collaborative filtering) and also recommends movies that share the same features with the movies that a users gave high ratings (i.e., content-based filtering) (Gomez-Uribe & Hunt, 2015).

2.1.4 Basic Problems of Recommender Systems

2.1.4.1 Cold start problem

The term cold start problem is derived from cars. In very cold weather, the engine of a car would have problems with starting up. But when it reaches its optimum operating temperature, it will run appropriately. For recommender system concept, the cold start refers to that the circumstances are not good enough for the engine to detect the best possible results (Gaspar, 2015).

Cold start problem is a problem when a new user or a new movie comes into the system. It is not possible to make suggestions because it is not known that what kind of films the new user will love and what kind of tracking behavior he/she will have. In this case, it is possible to make a suggestion by waiting the behavior of the new user in the system for the first week. It is also possible to suggest the most popular content or predicting the average rating of the target movie.

It is also a cold start problem that, when a new movie added to the system, which users it will be recommended to. The answer of this question would be the content-based filtering. First, metadata of new products can be used when creating recommendations, while user action is secondary for a certain period of time (Gaspar, 2015).

2.1.4.2 Sparcity problem

The aim of the collaborative filtering method is making recommendations based on users who have similar interests. However, because of users' lack of knowledge or incentives to rate items, ratings of user-item matrix is generally very sparse (up to 99%). Usually,

new users make a few or no ratings or new items receive only a few or no ratings. In this kind of situation, collaborative filtering can not provide effective recommendations, since users' preference is hard to obtain. Many algorithms have been developed for this problem, but these issues have not been well-addressed yet (Guo, 2012).

2.1.4.3 Scalability problem

Recommendation systems try to to produce suggestions about many products for many users. With the increase in the number of products and users, the scalability of the data becomes difficult. In neighborhood-based methods used in recommendation systems, users are represented by n-dimensional vectors and items are represented by m-dimensional vectors. In this case, calculating user to user and item to item similarities will bring a considerable amount of load on the system. In order to avoid this problem, matrix decomposition is used as a method in the recommendation systems (Koren, Bell & Volinsky, 2009). In the matrix decomposition method, if m is the total number of users, n is the the total number of items, each user and product is represented by $f << m,n$ sized vectors.

2.1.4.4 User privacy problem

One of the problems which affects recommendation systems is the privacy problem. System recommends items similar to that the users' already used. In this case, other items that the users may like can not be suggested. But, recommendation systems should suggest items both compatible with users' past behaviour and users may like in the future. Privacy problem can be solved with hybrid filtering methods. Content based system provides recommendation of items according to users's taste and collaborative filtering system makes recommendation using taste of similar users to the target user. Hybrid systems that use these two methods together is a solution to the privacy problem.

2.2 LITERATURE REVIEW

Choi, Ko and Han (2012) proposed an algorithm which improves results of the existing genre based recommender. The aim of this recommender is to solve cold-start and sparcity problems of the collaborative filtering approach. Since the genre information is determined by experts after the content of the movie completed, it is more reliable than the user ratings. First, genre correlations are calculated using genre combinations of each movie. Than, predicted rating is calculated using genre combination of all movies, user-preferred genres which the users request recommendation, average rating of each movie and genre correlations. In the previous method, when average rating of two movies are the same, the one which has more genres gets more points for prediction. New method prevented this mistake.

Uluyagmur (2012) proposed content-based, collaborative and hybrid recommendation methods and evaluated these algorithms. The author used actor, director, genre and some keywords received from the movie summaries for the content-based recommendation system. Each of these features are calculated seperately and their performances are measured. The method which uses director feature gives the best result. For the collaborative filtering method, users' ratings to the movies are used. Implicit ratings which are calculated by the percentage of the movie watched by that user are used since the study was done for the TV movies. Matrix factorization method was implemented as collaborative filtering method. Hybrid movie recommendation system combines recommendations of the content-based and collaborative filtering methods. Precision, recall, normalized precision and rating weighted normalized precision perfomance measures were used to evaluate the performance of the recommendation algorithms. According to this study, hybrid movie recommendation system gives the best results.

Ahn and Shi (2007) analyzed 5 types of cultural metadata such as user comments, plot outline, synopsis, plot keywords, and genres in IMDB system and detected which metadata enables more precise recommendation. They divided this metadata into 2 types which are text type and keyword type. Comments, plot outline and synopsis are text type

and plot keywords and genres are keyword type. Text type metada needs preprocessing and Natural Language Processing (NLP) before analyzing. But, keyword type metada do not need this kind of processes. The data is collected by web crawling. They created document vectors, detected TF-IDF of the metadata and used cosine similarity while calculating the similarities between the movies. User comments metadata has the highest precision according to the result of this study. The second successful metadata was the genres. Also, genres metadata enables to recommend more movies than the other metadata types. This study implies that how user comments are important as metadata for content based recommendation systems.

Mak, Koprinska and Poon (2003) compared the text based and feature based recommendations in their study. For the feature based recommendation, genre, director, leading actor and actress, awards won, awards nominated, country of origin features are used while synopsis of the movies were used for text based recommendation. The words in the synopsis of the movies were first separated into their roots, and the tf-idf vectors were generated and each movie was shown with a document vector. Inverse Document Frequency (IDF), Information Gain (IG) and Mutual Information (MI) methods are compared and best results are received with IDF method for eliminating stop words or words which has less information. In general, feature based method was working better for most of the users, but text based recommendation system outperformed feature based technique under particular conditions. For instance, text based classification is more successful when the number of ratings given by the user is higher than the number of words used to define documents.

3. DATA AND METHODOLOGY

The rating prediction performance of the content based and collaborative filtering models have been discovered and compared for a Turkish web site dataset in the scope of this study. In order to measure the prediction success of the content based filtering recommendation model, a project was implemented. For the collaborative filtering model, MyMediaLite open-source library was used.

3.1 CONTENT BASED FILTERING

Two different kinds of content based filtering methods were implemented and the prediction success of these methods were compared. These methods are plot decription based and metada and keywords based recommender methods. The project that was prepared for measuring the prediction success of these methods was implemented with C# programming language and entity framework was used for ORM (Object Relational Mapping). In order to collect the summary, genre, director and cast information of the movies, a web crawler was implemented using Selenium library. Also, user, movie and rating information was collected for creating the dataset that will be used in measurements. All the data was stored in MSSQL database. Tools, implementation details and methods that were used in the skeleton of this project will be explained in detail in this section.

3.1.1 Tools

3.1.1.1 Selenium

Selenium is a browser automation tool that allows creating and operating some test steps of websites automatically through a web browser. For example, correct and incorrect working scenarios of a website's user registration page can be tested with Selenium by automating the steps such as opening the user registration page, filling in the input fields

on the page and clicking the button. This makes easier to perform functional tests of the application. This way, different scenarios can be saved and instead of manually performing these scenarios, Selenium can do them later (Gojare, Joshi & Gaigaware, 2015).

Selenium is an open-source tool and there is no licensing cost involved, which is a major advantage over other testing tools. Test scripts can be written in several programming languages like Java, Python, C#, PHP, Ruby, Perl and .Net. Tests can be carried out in any of the Windows, Mac or Linux operating systems and using Mozilla Firefox, Internet Explorer, Google Chrome, Safari or Opera browsers. It can be integrated with TestNG and JUnit tools for managing test cases and generating reports. It can be integrated with Maven, Jenkins and Docker in order to make continuous testing (Jain & Kaluri, 2015).

Figure 3.1: Browsers supported by Selenium

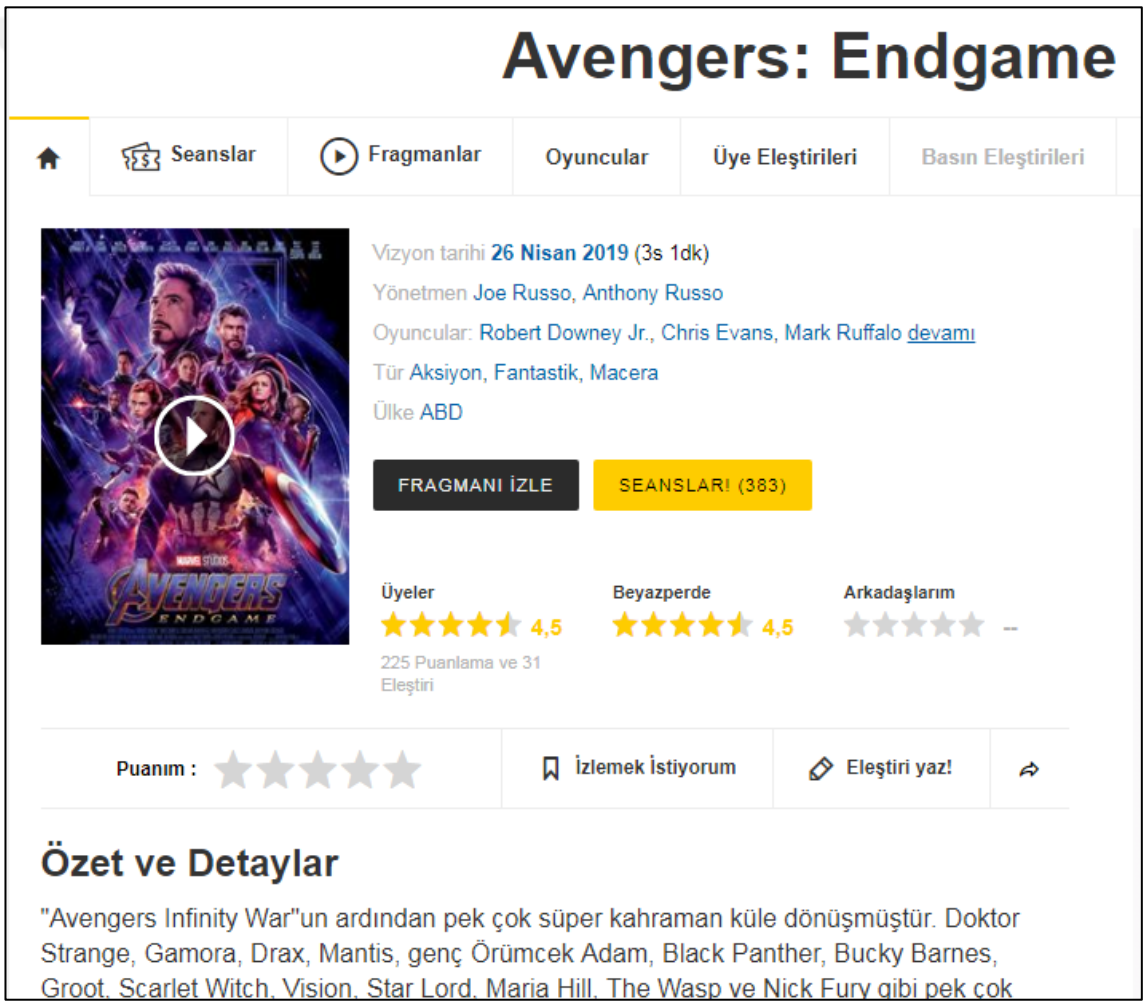
Source: Vardhan, 2019

Selenium is a suite of tools which consists of Selenium IDE (Integrated Development Environment), Selenium RC (Remote Control), Selenium Webdriver, and Selenium Grid. Selenium IDE is the simplest framework in the Selenium Suite. Scripts can be recorded and playbacked by this IDE. Simple scripts can be created using Selenium IDE. Selenium RC or Selenium WebDriver is needed to be used to write more advanced and robust test cases. Selenium WebDriver is a browser automation framework which accepts commands and sends them to a browser. It directly communicates and controls the browser. Selenium WebDriver supports Java, C#, PHP, Python, Perl and Ruby. Selenium

Grid supports distributed test execution which allows running the tests on different machines against different browsers in parallel (Rajkumar, 2018).

In this study, plot, genre, actors and director information as well as user, movie, rating data were collected by using Selenium WebDriver as the framework and Firefox as the browser. As seen in figure 3.2, plot description is received from *Özet and Detaylar* section. Cast information is received from *Oyuncular*, Director information is received from *Yönetmen* and genre information is received from *Tür* sections.

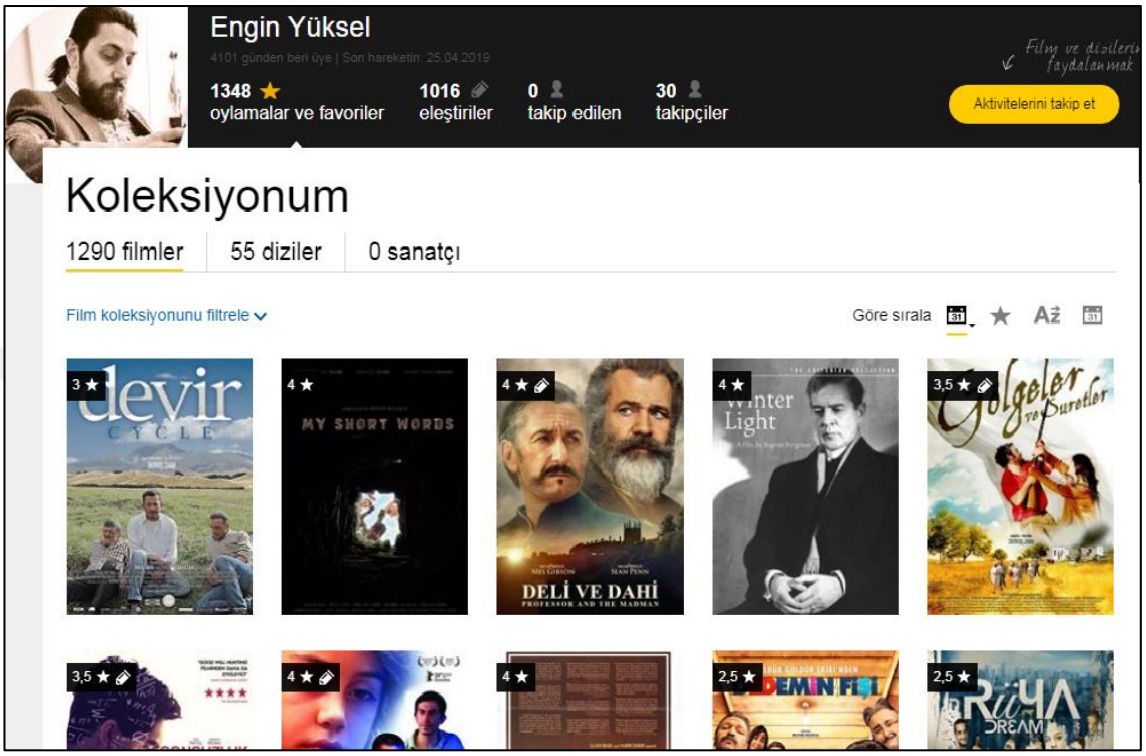
Figure 3.2: Beyazperde.com web site for plot description collection



User, movie, rating dataset is collected by clicking “üyeler” section which can be seen in the above figure as well. Under this section, users who rated the corresponding movie are

listed. Selenium WebDriver clicks all the users. When clicking each user, movies which the user is rated are listed as shown in Figure 3.3.

Figure 3.3: Beyazperde.com web site for user,movie,rating data collection



Selenium webdriver gets the rating data that exist on the movie picture with the system's user ID and movie ID. This process is executed for all movies and all users under that movies.

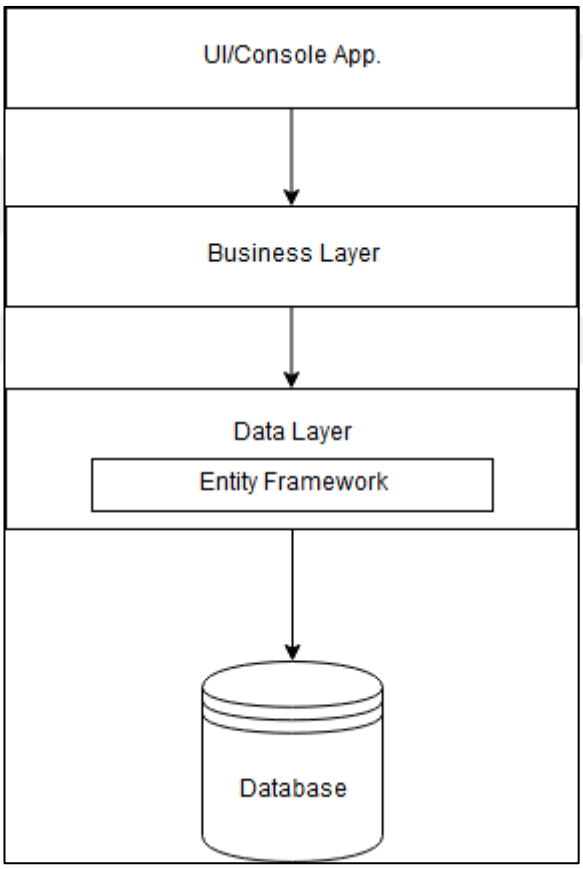
3.1.1.2 Entity Framework

Developers used to implement ADO.NET code or Enterprise Data Access Block to save or retrieve data from the database prior to .NET 3.5. First, a connection to the database was opened, than a dataset to fetch was created or the data was submitted to the database and the data from the dataset was converted to .NET objects. This was an ineligible and error prone process. Microsoft has provided an open-source ORM framework for .NET

applications called Entity Framework for automating all these database related activities (Adya, Blakeley, Melnik & Muralidhar, 2007).

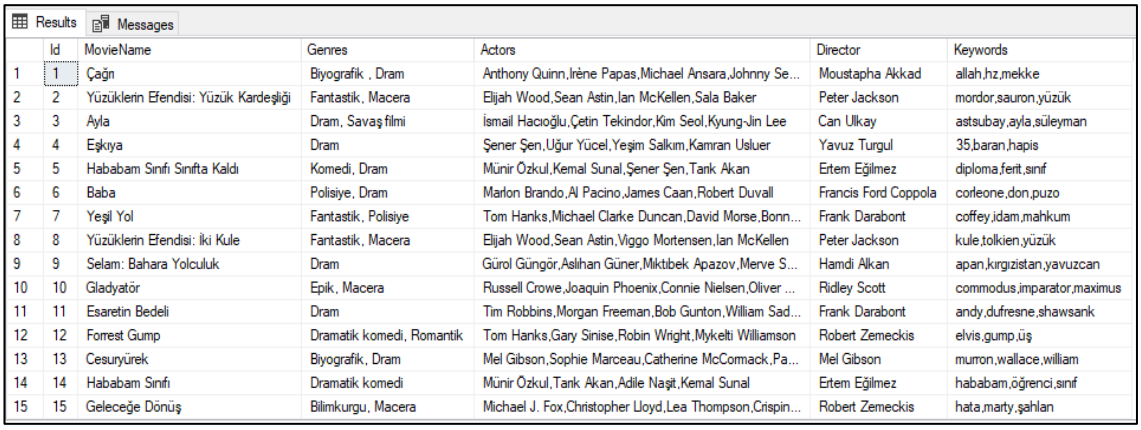
Entity Framework provides working at a higher level of abstraction without focusing on the related database tables and columns where this data is stored. With the Entity Framework, developers can create and maintain data-oriented applications with less code compared with traditional applications. Entity Framework layers are shown in Figure 3.4.

Figure 3.4: Entity Framework



In Figure 3.5 and Figure 3.6, the data that is gathered from the *www.beyazperde.com* website and inserted to the database by the help of entity framework can be seen.

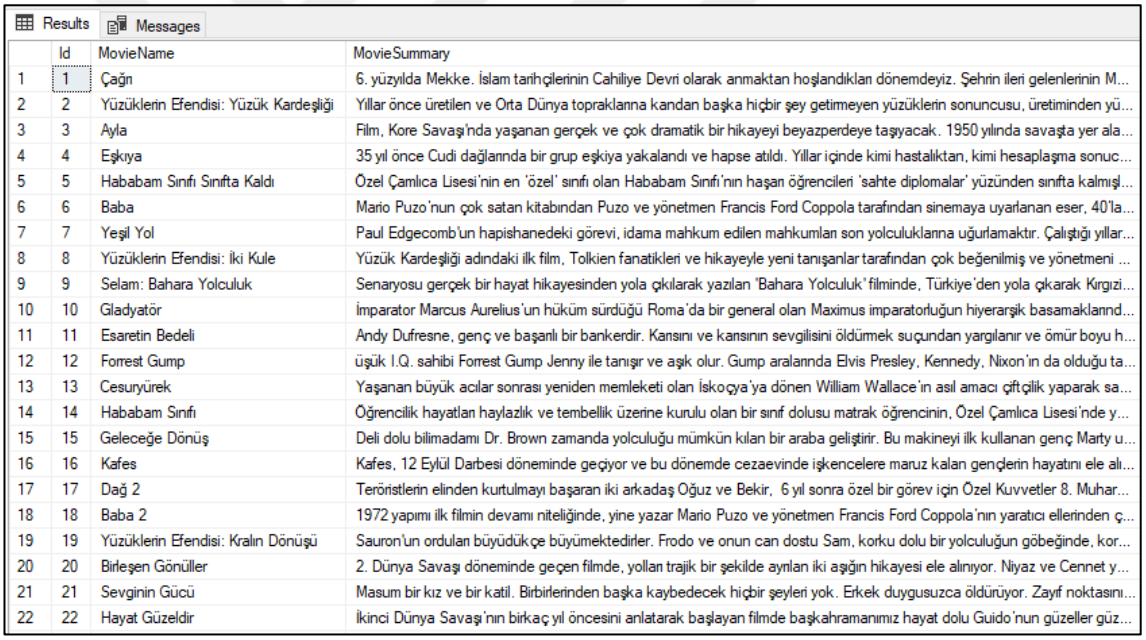
Figure 3.5: Metadata information for content based filtering in the database



Results | Messages

	Id	MovieName	Genres	Actors	Director	Keywords
1	1	Çağrı	Biyografik , Dram	Anthony Quinn,Irène Papas,Michael Ansara,Johnny Se...	Moustapha Akkad	allah,hz,mekke
2	2	Yüzüklerin Efendisi: Yüzük Kardeşliği	Fantastik, Macera	Elijah Wood,Sean Astin,Ian McKellen,Sala Baker	Peter Jackson	mordor,sauron,yüzük
3	3	Ayla	Dram, Savaş filmi	İsmail Hacıoğlu,Çetin Tekindor,Kim Seol,Kyung-Jin Lee	Can Ulkay	astsubay,ayla,süleyman
4	4	Eşkıya	Dram	Şener Şen,Uğur Yücel,Yeşim Salkım,Kamran Usluer	Yavuz Turgul	35,baran,hapis
5	5	Hababam Sınıfı Sınıfta Kaldı	Komedi, Dram	Münir Özkul,Kemal Sunal,Şener Şen,Tarık Akan	Ertem Eğilmez	diploma,ferit,sınıf
6	6	Baba	Polisiye, Dram	Marlon Brando,Al Pacino,James Caan,Robert Duvall	Francis Ford Coppola	corleone,don,puzo
7	7	Yeşil Yol	Fantastik, Polisiye	Tom Hanks,Michael Clarke Duncan,David Morse,Bonn...	Frank Darabont	coffey,idam,mahkum
8	8	Yüzüklerin Efendisi: İki Kule	Fantastik, Macera	Elijah Wood,Sean Astin,Viggo Mortensen,Ian McKellen	Peter Jackson	kule,tolkien,yüzük
9	9	Selam: Bahara Yolculuk	Dram	Gürol Güngör,Aslıhan Güner,Miktıbek Apazov,Merve S...	Hamdi Alkan	apan,kırgızistan,yavuzcan
10	10	Gladyatör	Epik, Macera	Russell Crowe,Joaquin Phoenix,Connie Nielsen,Oliver ...	Ridley Scott	commodus,imparator,maximus
11	11	Esaretin Bedeli	Dram	Tim Robbins,Morgan Freeman,Bob Gunton,William Sad...	Frank Darabont	andy,dufresne,shawsank
12	12	Forrest Gump	Dramatik komedi, Romantik	Tom Hanks,Gary Sinise,Robin Wright,Mykelti Williamson	Robert Zemeckis	elvis,gump,üş
13	13	Cesuryürek	Biyografik, Dram	Mel Gibson,Sophie Marceau,Catherine McCormack,Pa...	Mel Gibson	murron,wallace,william
14	14	Hababam Sınıfı	Dramatik komedi	Münir Özkul,Tarık Akan,Adile Naşit,Kemal Sunal	Ertem Eğilmez	hababam,öğrenci,sınıf
15	15	Geleceğe Dönüş	Bilimkurgu, Macera	Michael J. Fox,Christopher Lloyd,Lea Thompson,Crispin...	Robert Zemeckis	hata,marty,şahlan

Figure 3.6: Plot description information for content based filtering in the database



Results | Messages

	Id	MovieName	MovieSummary
1	1	Çağrı	6. yüzyılda Mekke. İslam tarihçilerinin Cahiliye Devri olarak anmaktan hoşlandıkları dönemdeyiz. Şehrin ileri gelenlerinin M...
2	2	Yüzüklerin Efendisi: Yüzük Kardeşliği	Yıllar önce üretilen ve Orta Dünya topraklarına kandan başka hiçbir şey getirmeyen yüzüklerin sonuncusu, üretiminden yü...
3	3	Ayla	Film, Kore Savaşı'nda yaşanan gerçek ve çok dramatik bir hikayeyi beyazperdeye taşıyacak. 1950 yılında savaşta yer ala...
4	4	Eşkıya	35 yıl önce Cudi dağlarında bir grup eşkiya yakalandı ve hapse atıldı. Yıllar içinde kimi hastalıktan, kimi hesaplaşma sonuc...
5	5	Hababam Sınıfı Sınıfta Kaldı	Özel Çamlıca Lisesi'nin en 'özel' sınıfı olan Hababam Sınıfı'nın haşarı öğrencileri 'sahte diplomalar' yüzünden sınıfta kalmışl...
6	6	Baba	Mario Puzo'nun çok satan kitabından Puzo ve yönetmen Francis Ford Coppola tarafından sinemaya uyarlanan eser, 40'la...
7	7	Yeşil Yol	Paul Edgecomb'un hapishanedeki görevi, idama mahkum edilen mahkumları son yolculuklarına uğurlamaktır. Çalıştığı yıllar...
8	8	Yüzüklerin Efendisi: İki Kule	Yüzük Kardeşliği adındaki ilk film, Tolkien fanatikleri ve hikayeyle yeni tanışanlar tarafından çok beğenilmiş ve yönetmeni ...
9	9	Selam: Bahara Yolculuk	Senaryosu gerçek bir hayat hikayesinden yola çıkılarak yazılan 'Bahara Yolculuk' filminde, Türkiye'den yola çıkarak Kırgızi...
10	10	Gladyatör	İmparator Marcus Aurelius'un hüküm sürdüğü Roma'da bir general olan Maximus imparatorluğun hiyerarşik basamaklarınd...
11	11	Esaretin Bedeli	Andy Dufresne, genç ve başarılı bir bankerdir. Karısını ve karısının sevgilisini öldürmek suçundan yargılanır ve ömür boyu h...
12	12	Forrest Gump	üşük I.Q. sahibi Forrest Gump Jenny ile tanışır ve aşık olur. Gump aralarında Elvis Presley, Kennedy, Nixon'ın da olduğu ta...
13	13	Cesuryürek	Yaşanan büyük acılar sonrası yeniden memleketi olan İskoçya'ya dönen William Wallace'in asıl amacı çiftçilik yaparak sa...
14	14	Hababam Sınıfı	Öğrencilik hayatları haylazlık ve tembellik üzerine kurulu olan bir sınıf dolusu matrak öğrencinin, Özel Çamlıca Lisesi'nde y...
15	15	Geleceğe Dönüş	Deli dolu bilimadamı Dr. Brown zamanda yolculuğu mümkün kılan bir araba geliştirir. Bu makineyi ilk kullanan genç Marty u...
16	16	Kafes	Kafes, 12 Eylül Darbesi döneminde geçiyor ve bu dönemde cezaevinde işkencelere maruz kalan gençlerin hayatını ele alı...
17	17	Dağ 2	Teröristlerin elinden kurtulmayı başaran iki arkadaş Oğuz ve Bekir, 6 yıl sonra özel bir görev için Özel Kuvvetler 8. Muhar...
18	18	Baba 2	1972 yapımı ilk filmin devamı niteliğinde, yine yazar Mario Puzo ve yönetmen Francis Ford Coppola'nın yaratıcı ellerinden ç...
19	19	Yüzüklerin Efendisi: Kralın Dönüşü	Sauron'un orduları büyüdükçe büyümektedirler. Frodo ve onun can dostu Sam, korku dolu bir yolculuğun göbeğinde, kor...
20	20	Birleşen Gönüller	2. Dünya Savaşı döneminde geçen filmde, yolları trajik bir şekilde ayrılan iki aşığın hikayesi ele alınıyor. Niyaz ve Cennet y...
21	21	Sevginin Gücü	Masum bir kız ve bir katil. Birbirlerinden başka kaybedecek hiçbir şeyleri yok. Erkek duygusuzca öldürüyor. Zayıf noktasını...
22	22	Hayat Güzeldir	İkinci Dünya Savaşı'nın birkaç yıl öncesini anlatarak başlayan filmde başkahramanımız hayat dolu Guido'nun güzeller güz...

4771 movies, 26638 users and 771274 user ,movie, rating data is received from the *www.beyazperde.com* website in the scope of this study.

3.1.1.3 Zemberek

Since Turkish is a agglutinative language, the language structure differs from other languages like extreme usage of affixes. This difference caused studies to be difficult, very limited and narrow level in the NLP (Natural Language Processing) field of the Turkish language. Because of the fundamental differences of agglutinative languages, it is very difficult to find the root of the word. The solution of the NLP problem, which is faced by Turkish language, is the creation of an education set by dividing the words into the roots (Değerli, 2012).

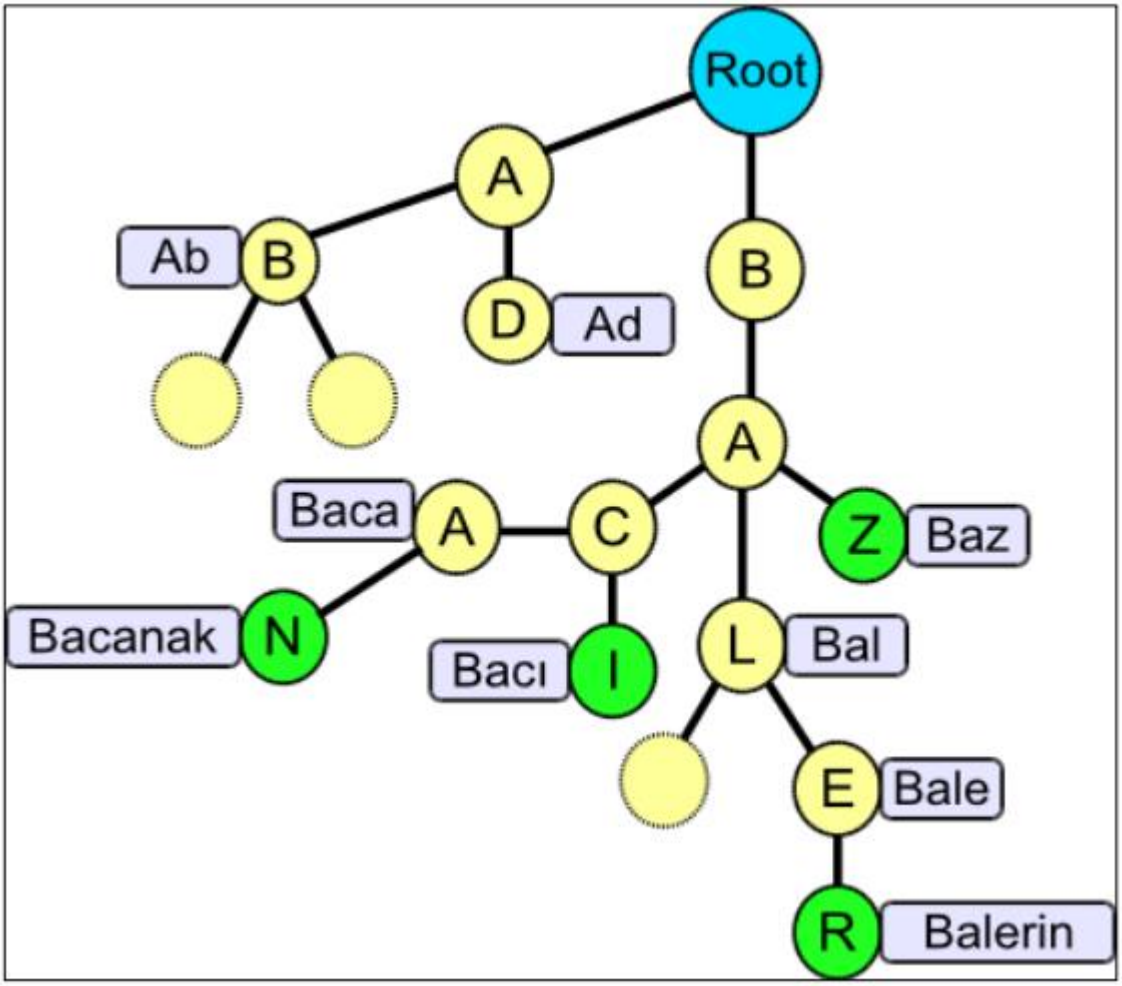
Zemberek started to be developed with Java for all neglected Turkish languages and especially for Turkish in 2004. It is an open source, platform independent and general purpose NLP library and a Libre Office extension. Zemberek offers a dictionary-based root finding method using the root and suffix dictionary files for analysis (Tunalı & Bilgin, 2012).

When the general structure of the Zemberek is investigated, it is divided into three parts as kernel, language implementations and data. In the kernel ,basic natural language processing algorithms, in the language implementations Turkish languages that are within the scope of the study and in the data, letters, roots and suffix data information is available (Eken, Ekinci & Sayar, 2014).

This library offers many NLP transactions like format analysis for the user, word based spelling, finding word suffix, spelling, spell checking, suggestion for incorrect words, incorrect coding, suffix separation, and word derivation (Tunalı & Bilgin, 2012).

Zemberek's root finding mechanism is used in this study. During the initialization of Zemberek, this library loads the binary root file first. Related special cases are attached to the root object when a root word is read. In order to provide fast access, the resulting object is stored into a special tree. Simplified structure of such a tree is shown in Figure 3.7 (Akın & Akın, 2016).

Figure 3.7: Root word tree



Source: Akın & Akın, 2016

3.1.2 Plot Description Based Recommender

3.1.2.1 TF-IDF method

The aim of this recommender is making rating prediction using plot description of the movies. First, item to item similarity matrix should be created. For this reason, punctuation marks are extracted from plot descriptions and all of the words are sent to Zemberek tool in order to find their roots. Usually, a couple of results are received for one word as root. First one is selected, others are ignored even if there is a possibility that one of the others would be correct. Because, there is no way to decide which one is correct automatically. If the tool can not find any root candidate, then the word itself is used.

Figure 3.8: Flow chart of plot description based recommender application

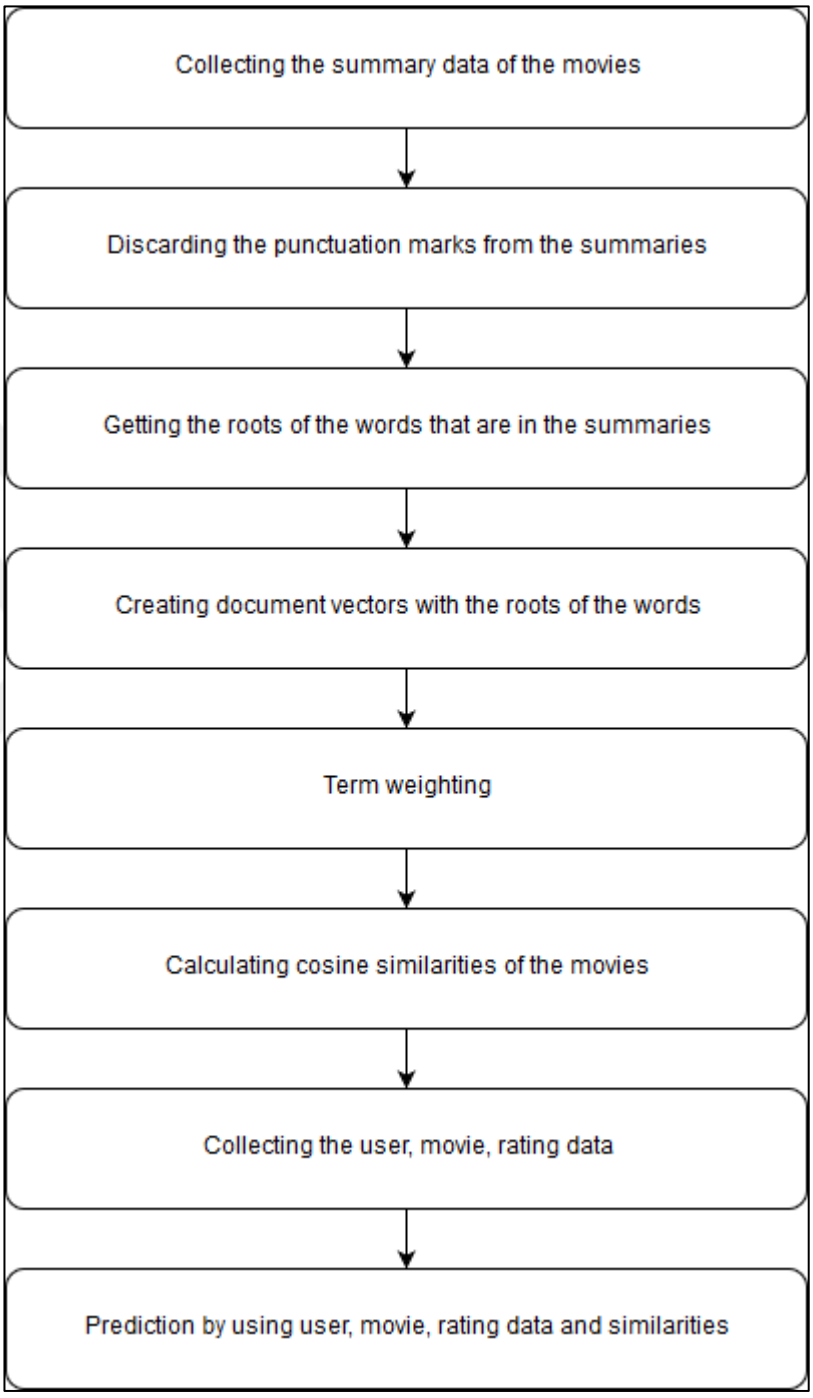
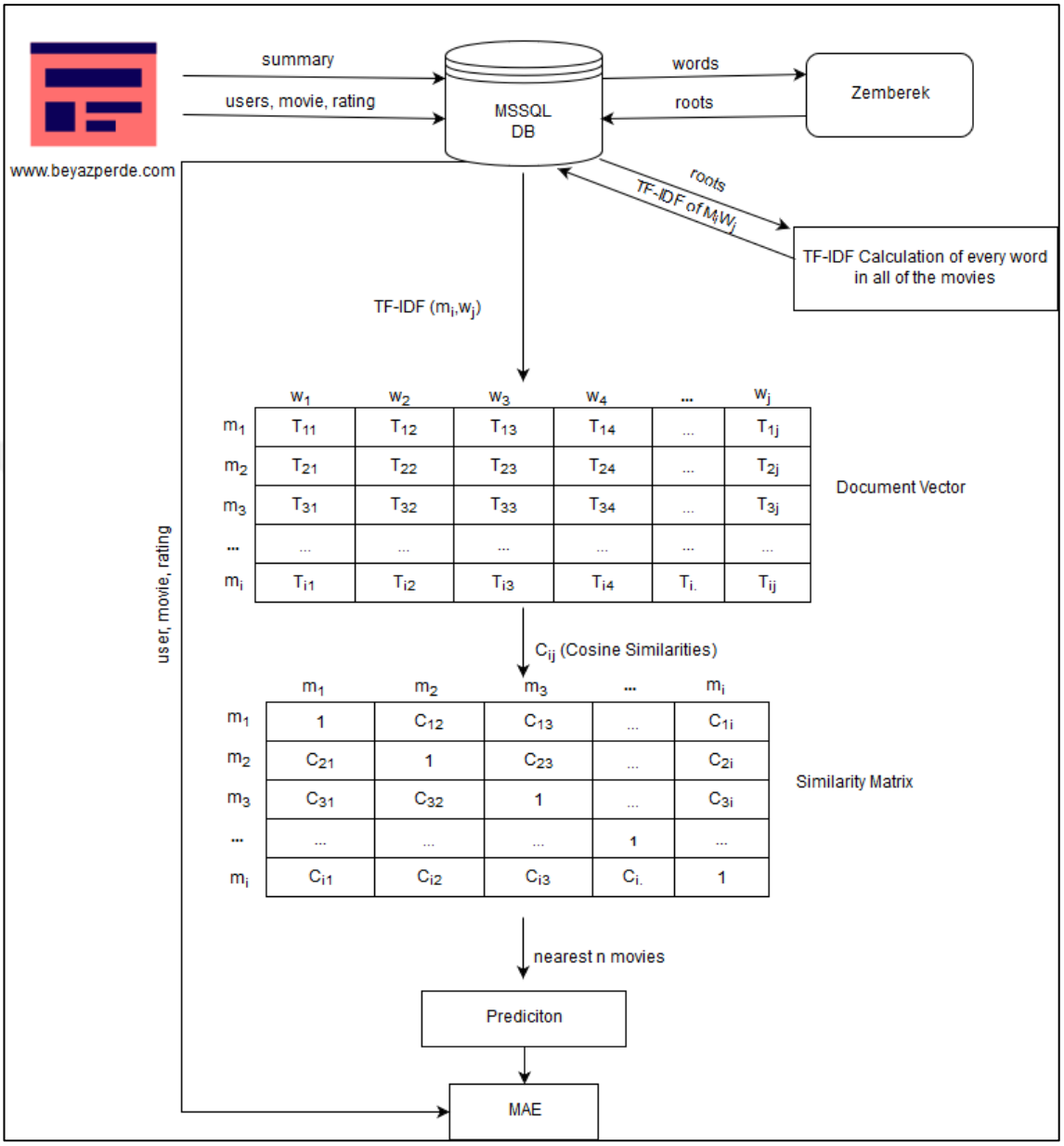


Figure 3.9: Implementation model of plot description based recommender application



After roots of all of the words are found, a document vector model with all of the roots are created for all of the movies. This is also called corpus or bag of words. In order to prepare this model, a two dimentional matrix is created. Rows of this matrix represent movies as m*i* and columns of this matrix represents all of the root words of all of the movies as w*j*.

TF-IDF values for all the root words are calculated for all the movies and they are written in the appropriate cells as Tij. TF value is the ratio of how many times a word exist in a description to total number of words that description has. IDF value is the ratio of total number of movie descriptions to how many movie descriptions have that word. TF value explains how important that word to that plot description. IDF value is used for filtering the words which do not contain much information for that movie description which are generally stop words (e.g. and, the, that, like, ...). Formula for TF-IDF calculation can be found on equation (2.2).

After the document vector is filled with TF-IDF values, item to item similarity matrix can be created. This is a two dimentional matrix which every row and column refers to a movie. Cosinus similarities are calculated based on formula 2.3 using the document vectors and similarity matrix is filled with these values. Once the similarity matrix is filled, correlation of all movies are known. Diagonal of the similarity matrix is 1 since the similarity between the same movies are 1.

Predictions are calculated based on the item to item similarity matrix. While making rating prediction, calculations are made based on a coefficient 10 to 100 increased by 10. For example, if the coefficient is 10, users who rated at least 10 movies are taken into account. Also, in the similarity matrix, most similar 10 movies to the target movie are taken into account by selecting the greatest 10 correlations in the matrix.

Weighted rating is used while calculation. For instance, if the system is trying to predict the rating for user a and movie i, while making calculation for coefficient 10, it receives the greatest 10 values in similarity matrix in movie i row except the value 1 since it is movie i itself. What ratings user a gave to the 10 most similar movies are found out and those ratings are multiplied by their correlation values. Following simple weighted average formula is used while predicting the rating:

$$p_{a,i} = \frac{\sum_{j \in K} r_{a,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|} \tag{3.1}$$

In the formula above, *K* refers to the neighborhood of most similar items rated by active user a and *w(i,j)* is the similarity between items *i* and *j*. The result of this calculation gives the predicted rating.

After making the rating prediction, it can be measured how successful this prediction is since the actual rating is known. Mean Absolute Error (MAE) is a metric which can be used for this reason. It refers to average value of absolute differences between the actual and the predicted values. All the individual differences are weighted equally while calculating MAE.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right| \tag{3.2}$$

In this study, MAE is used to measure accuracy for our content based filtering model.

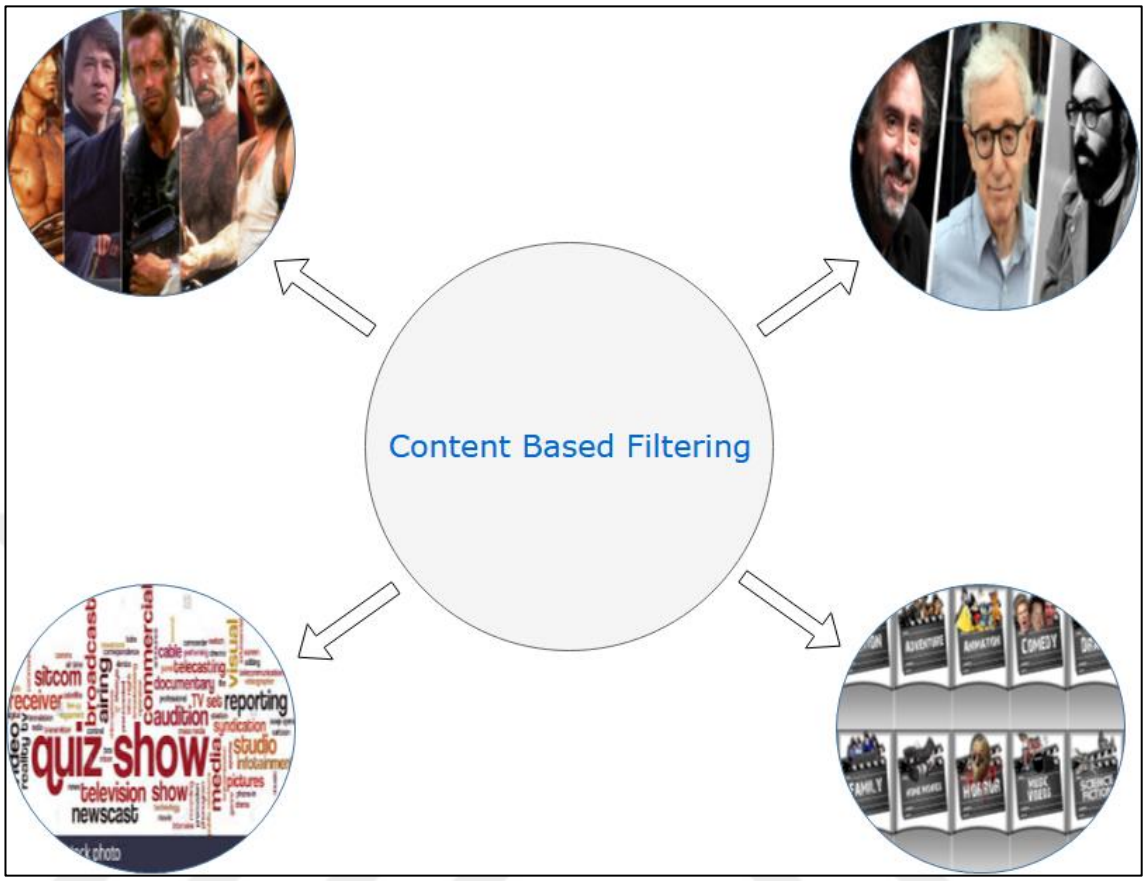
3.1.2.1 COUNT method

The only difference of count method from TF-IDF method is, total number of existence of a word in a movie description is filled to document vectors instead of TF-IDF values. Accuracy performance of this method is expected to be lower than TF-IDF method since the words which do not contain much information are not down weighted.

3.1.3 Metadata And Keywords Based Recommender

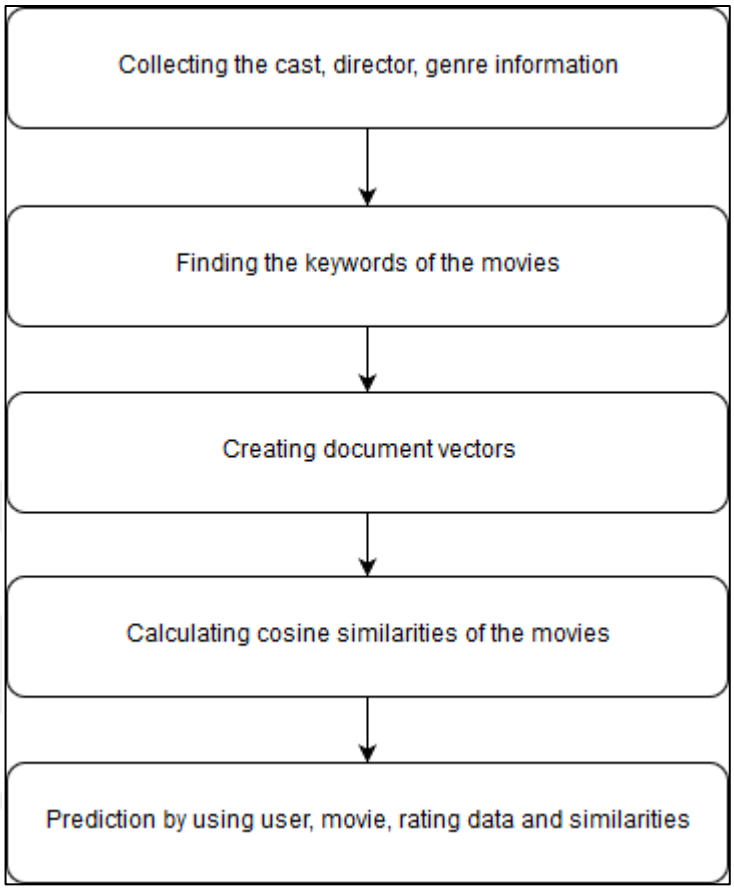
This type of recommender systems use actor/actress, director, genre and keyword information as content as shown in Figure 3.10.

Figure 3.10: Metadata and keywords content



Cast, director and genre information which is also called metadata is received from the web site by Selenium tool. Keywords are obtained from plot descriptions. Top 3 words which has the highest TF-IDF value is considered as keywords for each movie. Flow chart of metadata and keywords based recommender system application is shown in Figure 3.11.

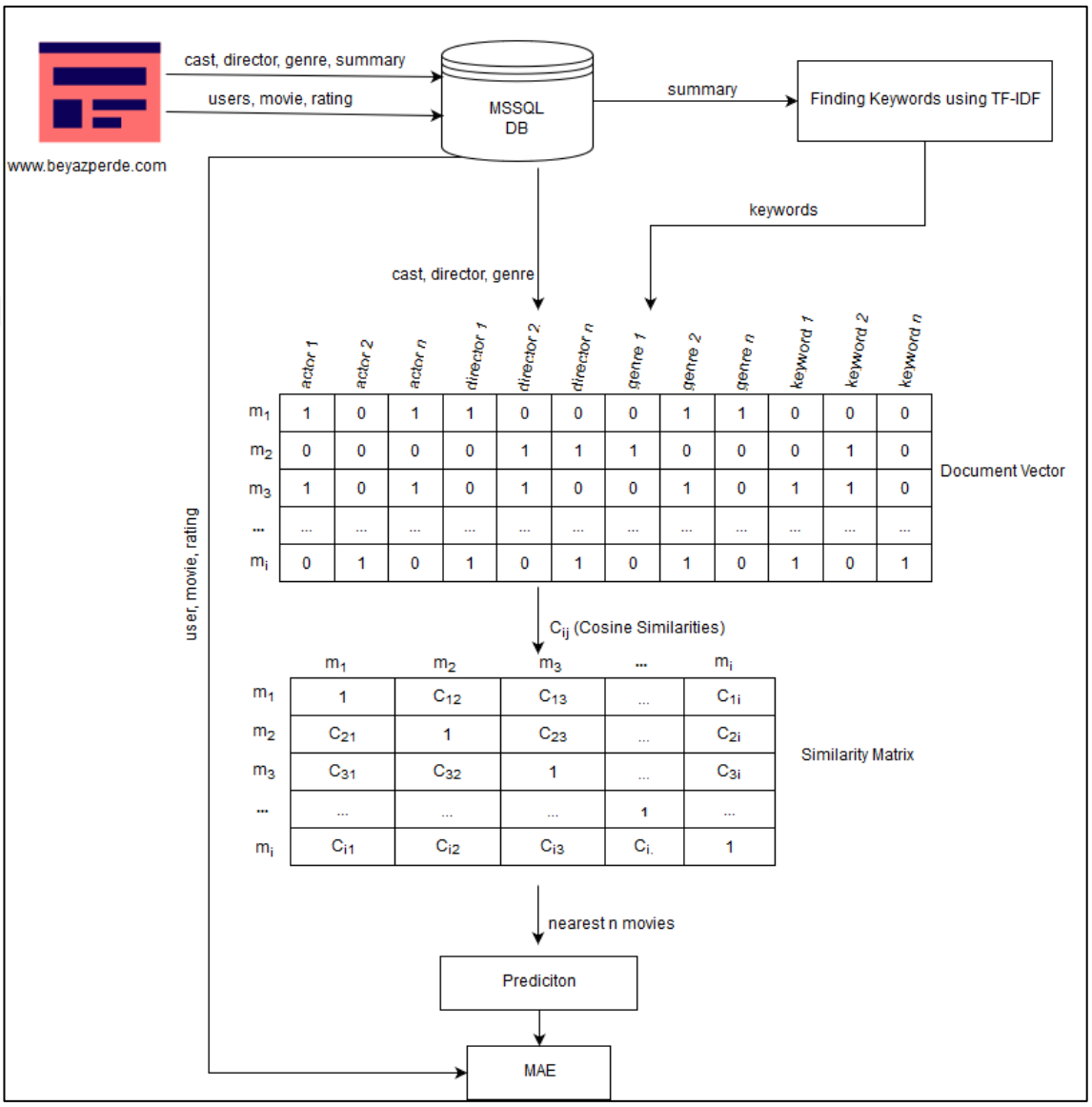
Figure 3.11: Flow chart of metadata and keywords based recommender system



After the keywords are extracted by TF-IDF method, document vectors are created. Document vectors are matrices that their rows refer to movies and their columns refer to metadata and keywords. There is a column for each metadata and keyword which exists in all of the movies. If the corresponding metadata or keyword exist in a specific movie, than the intersecting cell will have the value 1, otherwise it will have the value 0. For instance, for row of the movie *Hababam Sınıfı*, there will be value 1 under keyword *öğrenci* column, but for another movie which does not have the word *öğrenci* in any of its metada or keywords, its cell under the *öğrenci* column will be 0. Since the cast, directors, genre or keywords are unique values for each movie, in another words metadata *Kemal Sunal* can exist just once in the movie *Hababam Sınıfı*, cells are just filled with 1 or 0. On the other hand, TF-IDF values are not calculated since down weighting the

presence of an actor/director if he or she has acted or directed in relatively more movies does not make sense.

Figure 3.12: Implementation model of metada and keywords based recommender application



After the document vector is created and filled with appropriate values, item to item similarity matrix is created. Similarities between the items are calculated by using this document vector and cosinus similarity formula which can be seen as equation 2.3. Ratings are predicted by getting the top n similar items to the target item. Active user's ratings to these top n similar movies are taken into account by using equation 3.1 while

calculating predicted rating. When the ratings are predicted, MAE is calculated in order to measure the accuracy performance of this method.

3.2 COLLABORATIVE FILTERING

While measuring the accuracy performance of the collaborative filtering methods, MyMediaLite recommender system library is used in this study.

3.2.1 MyMediaLite Recommender System Library

MyMediaLite is a fast and open-source library for recommendation system algorithms which is built for recommender system researchers and practitioners. Rating prediction and item prediction from positive-only implicit feedback collaborative filtering scenarios are adressed by this library. Positive-only implicit feedback includes clicks or purchase actions. A common API and efficient data structures were used while implementing the algoritms. The algorithms offered by the MyMediaLite library are state-of-the-art algorithms for both rating prediction and item prediction and deep knowledge of programming is not required to use. So, recommendation system researchers are relieved from implementing the existing methods which are already in the MyMediaLite recommendation system library and they can easily compare the performance of their newly designed methods with the existing methods. MyMediaLite was built by using C# programming language and it runs on the .NET platform. It also can be called from the other programming languages like Python and Ruby (Gantner, Rendle, Freudenthaler & Thieme, 2011).

MyMediaLite just needs to be downloaded in order to be used. It is used through a command line program which exists in any operating system. Recommender systems are evaluated by this framework with criterias like Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

In this study, rating prediction algorithms are used for accuracy performance comparision of the recommender system methods.

Slope One and *BiPolar Slope One* algorithms are known as the simplest way of non-trivial item based collaborative filtering. This simplicity makes them to be implemented easily. Also, their accuracy is not lower than more complicated algorithms. They are also used in improvement of the other algorithms (Lemire & Maclachlan, 2005).

Global Average model computes the average value of all ratings, *Item Average* model computes the average rating value of an item and *User Average* model computes the average rating value of a user in order to make predictions. *User Item Baseline* model sums the bias value of a user, bias value of an item and global average value (Carrion, 2016).

Co-Clustering algorithm is also known as BiClustering and simultaneous clustering of the rows and columns of a matrix. In general clustering methods, an object is a member of a group which resemble its entity type. In CoClustering, two types of entities are co-grouped simultaneusly based on similarity of their pairwise interactions (Reshef, 2015).

Random algorithm is used for measuring if the predictions are made randomly, not based on any similarity model.

In the usual log-linear model, all variables can be directly observed. However, sometimes a variable itself can not be directly measured and several latent indicators need to be used to measure a concept. For factor analysis, some indicators can be used to measure a latent variable. *Latent Feature Log Linear Model* investigates the relationships between the manifest and the latent variables (Vermunt, 1996).

Biased Matrix Factorization (BMF) algorithm uses explicit bias for items and users learning by stochastic gradient descent (Barjasteh, 2016).

SVD Plus Plus (SVD++) is a matrix factorization algorithm which uses what users have rated and profiles users and items directly. This model uses both implicit and explicit information when regular SVD model uses only explicit information (Barjasteh, 2016).

Sigmoid SVD Plus Plus (SSVD++) model is a version of SVD++ which uses sigmoid function (Barjasteh, 2016).

Sigmoid User Asymmetric Factor Model (SU-AFM) model represents the items with regards to those users that rated them (Barjasteh, 2016).

Sigmoid Item Asymmetric Factor Model (SI-AFM) model represents the users with regards to the items they rated (Barjasteh, 2016).

Sigmoid Combined Asymmetric Factor Model (SC-AFM) model represents the items with regards to those users that rated them and the users with regards to the items they rated (Barjasteh, 2016).

Matrix Factorization, ItemKNN and UserKNN algorithms explained in section 2 in detail.

4. EXPERIMENTAL RESULTS

In this section, experimental results are explained for both content based filtering and collaborative filtering techniques and their rating prediction accuracies are measured and compared. The data that is used in the test is gathered from *www.beyazperde.com* website by implementing a web crawler using Selenium Webdriver.

Tests are performed for 3 types of content based filtering methods which are TF-IDF, count and metadata and keywords based methods. For count and TF-IDF methods, system model which can be seen in Figure 3.9 is applied. For metadata and keywords based methods, system model which can be seen in Figure 3.12 is applied. Following tables 4.1, 4.2, 4.3 and Figures 4.1, 4.2, 4.3 show the accuracy performance of these models. Also, in Figure 4.4 comparison of the accuracy performances of these methods can be seen.

Table 4.1: MAE results for content based methods

Coefficient	Count	TF-IDF	Metadata
10	0,773	0,751	0,729
20	0,764	0,735	0,717
30	0,755	0,749	0,706
40	0,776	0,739	0,703
50	0,762	0,724	0,704
60	0,753	0,724	0,688
70	0,745	0,718	0,673
80	0,743	0,687	0,673
90	0,720	0,664	0,669
100	0,716	0,671	0,663

Figure 4.1: MAE results for count method

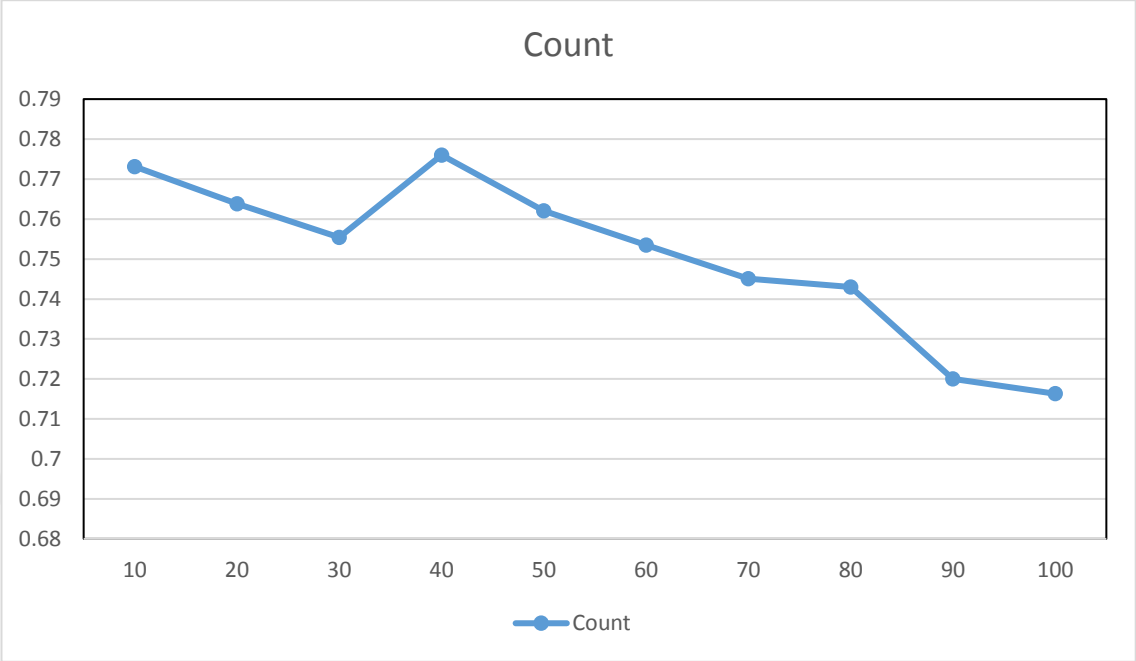


Figure 4.2: MAE results for TF-IDF method

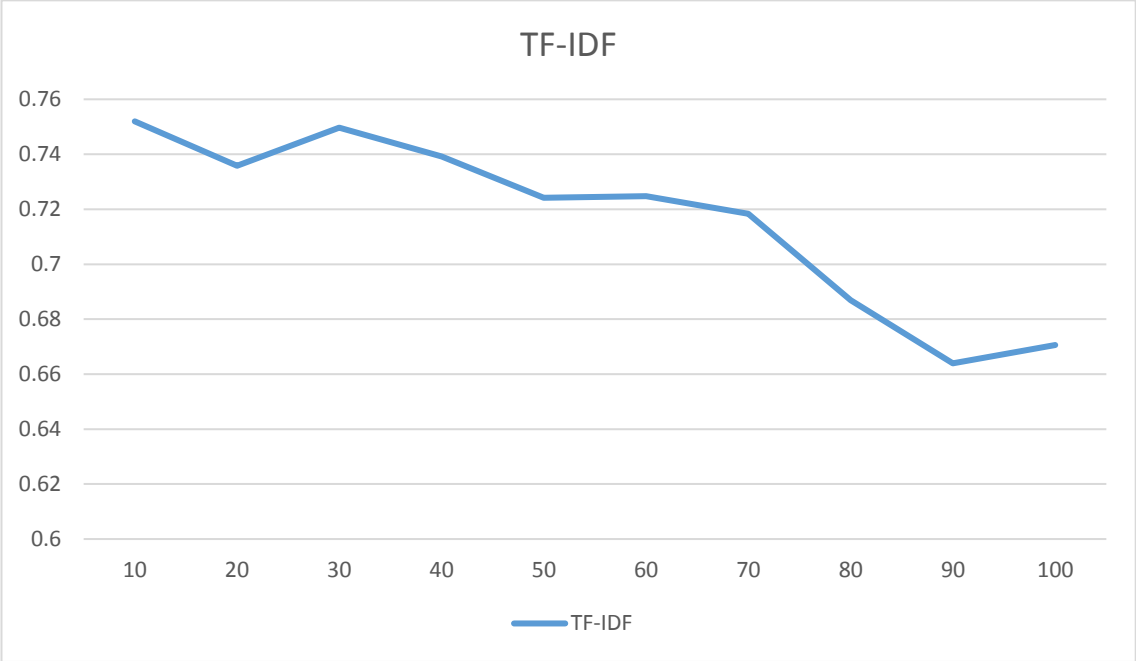


Figure 4.3: MAE results for metadata and keyword based method

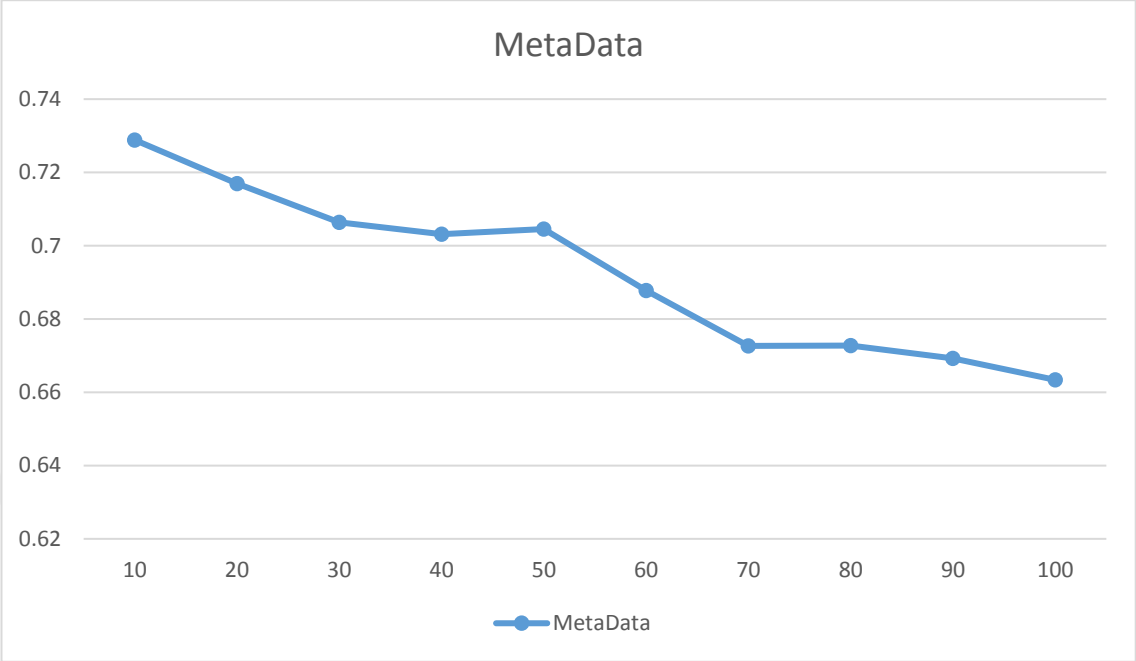
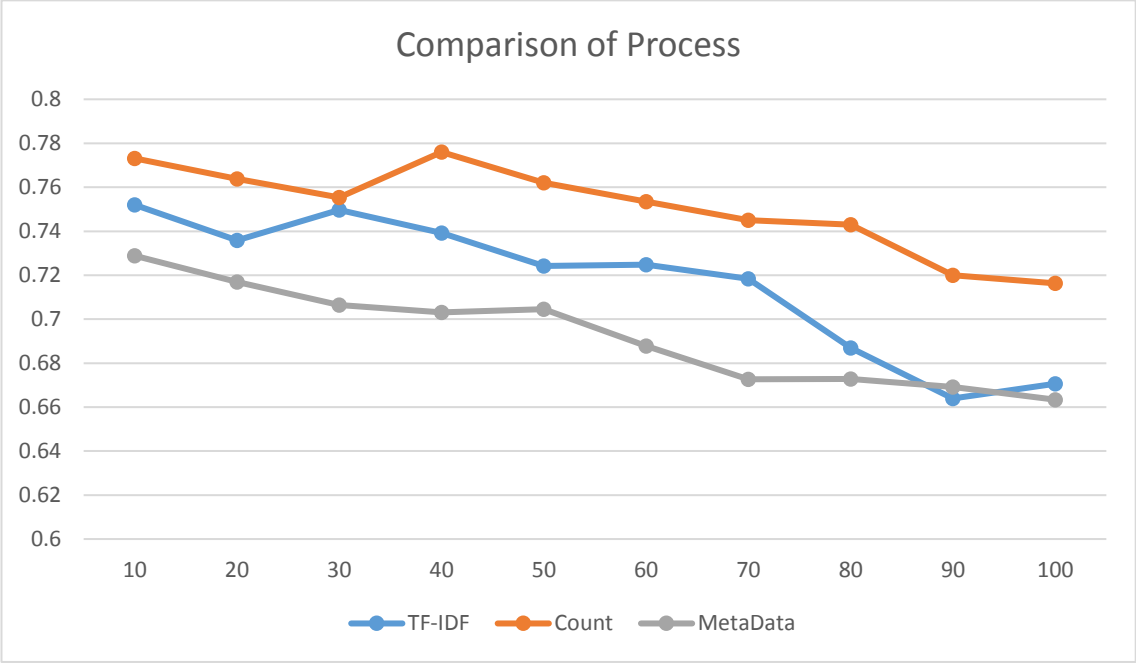


Figure 4.4: Accuracy comparision of 3 types of content based filtering methods



Average MAE results for these content based methods are shown in the table 4.2.

Table 4.2: Average MAE results for content based methods

Average MAE Results	
Count	0,751
TF-IDF	0,716
Metadata	0,692

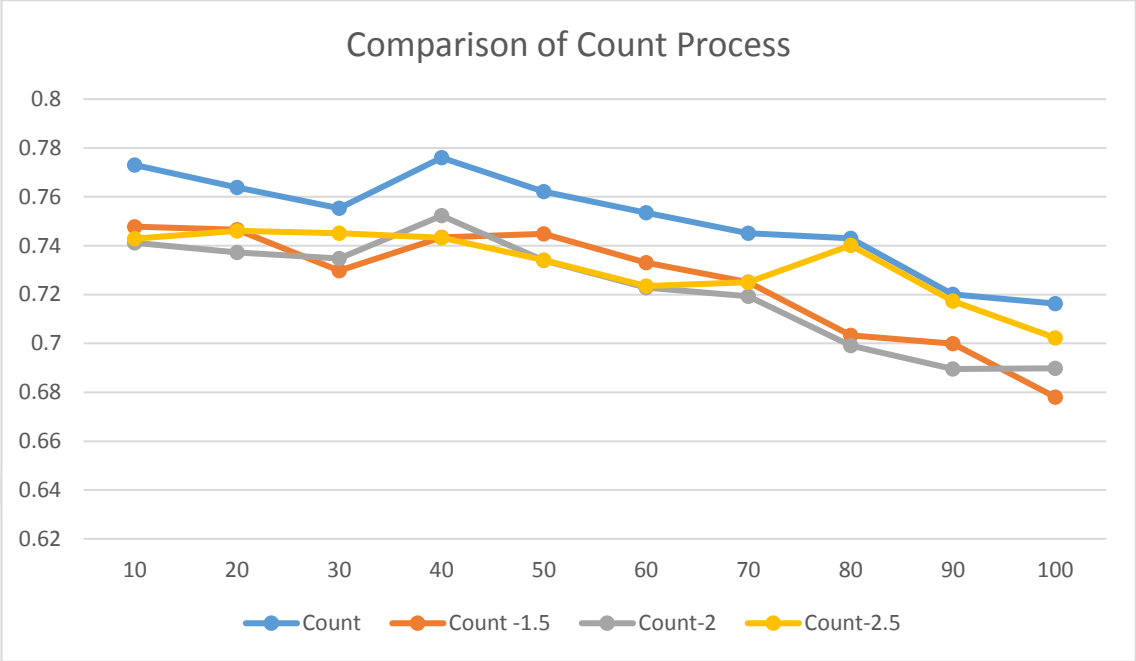
According to the test results, accuracy of TF-IDF method is 4.6% greater than the count method and accuracy of metadata and keywords method is 3.3% greater than the plot description based TF-IDF method.

Also, another technique is tested for count method. The words whose IDF values are under a specific treshold are discarded from calculations in order to understand if the words which do not contain much information affect the results. This test was performed for tresholds 1.5, 2 and 2.5. Best results are received for thresholds 1,5 and 2. It means that the words whose IDF values under 1,5 does not contain enough information and the words whose IDF values over 2 contain important information and affect the calculations in a bad way if they are discarded. Test results can be seen from the below table:

Table 4.3: MAE results for count method for different thresholds

Coefficient	No Threshold	Threshold 1.5	Threshold 2	Threshold 2.5
10	0,773	0,748	0,741	0,743
20	0,764	0,746	0,737	0,746
30	0,755	0,730	0,735	0,745
40	0,776	0,743	0,752	0,743
50	0,762	0,745	0,734	0,734
60	0,753	0,733	0,723	0,723
70	0,745	0,725	0,719	0,725
80	0,743	0,703	0,670	0,740
90	0,720	0,670	0,690	0,717
100	0,716	0,678	0,690	0,702

Figure 4.5: Accuracy comparision of count method for different thresholds



Average MAE results for different thresholds of count method can be seen in table 4.4.

Table 4.4: Average MAE results for count method for different thresholds

Average MAE Results for Count Methods	
Method	**Average MAE**
No Threshold	7,508
Threshold 1.5	7,252
Threshold 2	7,220
Threshold 2.5	7,320

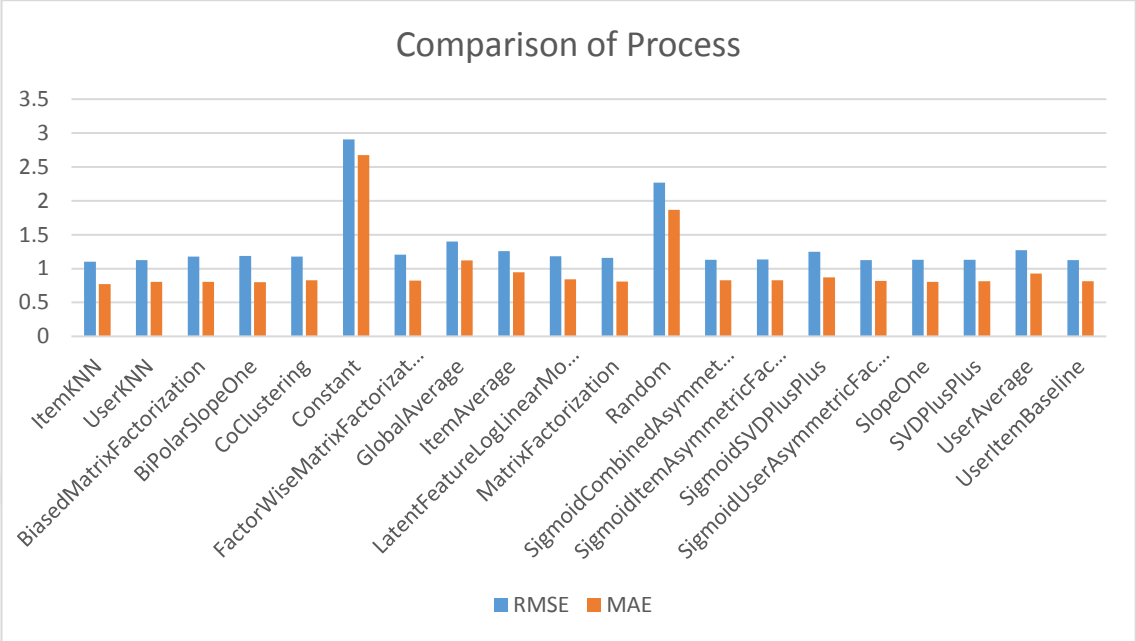
According to the above table, best result is received with threshold 2 for the count method. Its performance is % 0.4 better than threshold 1.5 and threshold 1.5 accuracy performance is % 0.9 better than threshold 2.5. The method applied with no threshold has worst performance which is % 1.8 worse than the method applied with threshold 2.5.

Tests for collaborative filtering methods are performed via MyMediaLite tool for several collaborative filtering methods. Best result is received with the item based collaborative filtering technique. MAE results are shown in table 4.5 and figure 4.6.

Table 4.5: Test results for collaborative filtering methods

Process	RMSE	MAE
ItemKNN	1,103	0,7711
UserKNN	1,124	0,805
BiasedMatrixFactorization	1,176	0,804
BiPolarSlopeOne	1,185	0,800
CoClustering	1,178	0,827
Constant	2,907	2,674
FactorWiseMatrixFactorization	1,204	0,823
GlobalAverage	1,398	1,121
ItemAverage	1,257	0,944
LatentFeatureLogLinearModel	1,180	0,841
MatrixFactorization	1,161	0,807
Random	2,270	1,865
SigmoidCombinedAsymmetricFactorModel	1,131	0,828
SigmoidItemAsymmetricFactorModel	1,135	0,828
SigmoidSVDPlusPlus	1,250	0,871
SigmoidUserAsymmetricFactorModel	1,124	0,816
SlopeOne	1,131	0,803
SVDPlusPlus	1,130	0,815
UserAverage	1,274	0,926
UserItemBaseline	1,127	0,812

Figure 4.6: Test results for collaborative filtering methods



Best results are received with metadata and keywords based recommender with average MAE 0.69 when comparing all of the results including both content based filtering and collaborative filtering techniques for *www.beyazperde.com* dataset.

5. CONCLUSION

In the scope of this study, three types of content based filtering methods are implemented and their success on rating prediction is measured for Turkish language. Two of these methods are based on plot description of the movies which are count and TF-IDF methods. The other method is called metadata and keywords technique and it uses actor/actress, director, genre and keyword information of the movies while calculating their similarities. All the required data is gathered from *www.beyazperde.com* website by implementing a web crawler. Also, accuracy success of the content based filtering methods are compared with several collaborative filtering methods. Tests of the collaborative filtering methods are performed via MyMediaLite tool using the same dataset with the content based methods. Best results are received with metadata and keywords method of the content based filtering technique.

For the future work, this study can be performed for data from another website in order to provide external validity. Also, hybrid filtering method which combines content based filtering and collaborative filtering methods can be applied with the same data and dataset.

REFERENCES

Books

Ahn S., Shi CK., 2009. Exploring Movie Recommendation System Using Cultural Metadata. In: Pan Z., Cheok A.D., Müller W., Rhalibi A.E. (eds) Transactions on Edutainment II. Lecture Notes in Computer Science, vol 5660. Springer, Berlin, Heidelberg

Burke R. (2007) Hybrid Web Recommender Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg

Golub, G.H., Reinsch, C. (1971) Singular Value Decomposition and Least Squares Solutions. In: Bauer F.L. (eds) Linear Algebra. Handbook for Automatic Computation, vol 2. Springer, Berlin, Heidelberg

Guo, G., 2012. Resolving Data Sparsity and Cold Start in Recommender Systems. In: Masthoff J., Mobasher B., Desmarais M.C., Nkambou R. (eds) User Modeling, Adaptation, and Personalization. UMAP 2012. Lecture Notes in Computer Science, vol 7379. Springer, Berlin, Heidelberg

Vermunt, J. K. (1996). Causal log-linear modeling with latent variables and missing data. Analysis of change: Advanced techniques in panel data analysis, 35-60.

Periodicals

Çano, E. & Morisio, M., 2017. Hybrid Recommender Systems: A Systematic Literature Review. *Intelligent Data Analysis*. 21. 1487-1524. 10.3233/IDA-163209.

Choi, S. M., Ko, S. K., & Han, Y. S., 2012. A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications*, ***39***(9), 8079-8085.

Eken, S., Ekinci, E., & Sayar, A., 2014. XML anahtar kelimeleri yardımıyla türkçe aritmetik problemlerin anlaşılması ve çözülmesi. *Düzce University Science and Technology Magazine*, (2), 48-55.

Gojare, S., Joshi, R., & Gaigaware, D., 2015. Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, *50*, 341-346.

Gomez-Uribe, C.A. & Hunt, N. (28 December 2015). The Netflix Recommender System. *ACM Transactions on Management Information Systems*. **6** (4): 1–19.

Jain, C. R., & Kaluri, R. (April, 2015). Design of automation scripts execution application for selenium webdriver and test NG framework. *ARPN Journal of Engineering and Applied Sciences*, *10*, 2440-2445.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix Factorization Techniques for Recommender Systems. *Computer*, Volume 42, Issue 8, 30-37.

Other Publications

Adya, A., Blakeley, J. A., Melnik, S., & Muralidhar, S. (2007, June). Anatomy of the ado.net entity framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (pp. 877-888). ACM.

Akın, A. A., & Akın, M. D. (2007). Zemberek, an open source nlp framework for turkic languages. *Structure*, *10*, 1-5.

Barjasteh, I. (2016). Matrix completion with side Information for effective recommendation. *Unpublished doctoral dissertation*. Michigan State University.

Carrion, S., 2016, baselines [online]. Available at: https://orange3-recommendation.readthedocs.io/en/latest/widgets/baselines.html [accessed 15 April 2019]

Christakou, C. & Stafylopatis, A., 2005. A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools - IJAIT*. 16. 500- 505. 10.1109/ISDA.2005.9.

Değerli, O. (2012). Naive Bayes Yöntemi ile Blog İçeriklerinin Sınıflandırılması. *Unpublished master's thesis*. Gazi University, Ankara,Turkey.

Do, T.M.P, Nguyen, D.V. & Nguyen, L., 2010. Model-based Approach for Collaborative Filtering, *The 6th International Conference on Information Technology for Education* (IT@EDU2010) Ho Chi Minh city, Vietnam

Durukan, F. , 2018, Öneri Sistemleri [online]. Available at: http://fatihdurukan.com/oneri-sistemleri/ [accessed 22 April 2019]

Gantner, Z., Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. , 2011. MyMediaLite: a free recommender system library. *Paper presented at the Proceedings of the fifth ACM conference on Recommender systems*, Chicago, Illinois, USA.

Gaspar, H., 2015. The Cold Start Problem for Recommender Systems [online]. Available at: https://www.yuspify.com/blog/cold-start-problem-recommender-systems/ [accessed 27 April 2019]

Grimaldi, E., 2018. How to build a content-based movie recommender system with Natural Language Processing [online]. Available at: https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie-recommender-with-natural-language-processing-25ad400eb243 [accessed 23 April 2019]

Guan, X., Li, C. T., & Guan, Y. (2016, April). Enhanced SVD for collaborative filtering. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 503-514). Springer, Cham.

Huang, S., 2018. Introduction to Recommender System. Part 1 (Collaborative Filtering, Singular Value Decomposition) [online]. Available at: https://hackernoon.com/introduction-to-recommender-system-part-1-collaborative-filtering-singular-value-decomposition-44c9659c5e75?gi=2cb94cbf8ac9 [accessed 26 April 2019]

Lemire, D., & Maclachlan, A. (2005, April). Slope one predictors for online rating-based collaborative filtering. *In Proceedings of the 2005 SIAM International Conference on Data Mining (pp. 471-475)*. Society for Industrial and Applied Mathematics.

Mak, H., Koprinska, I., Poon, J., 2003. INTIMATE: a Web-based movie recommender using text categorization, *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, Halifax, NS, Canada, pp. 602-605.

Rajkumar, S.M., 2018. Selenium WebDriver Architecture [online]. Available at: https://www.softwaretestingmaterial.com/selenium-webdriver-architecture/ [accessed 20 April 2019]

Reshef, R., 2015. How to build a recommendation engine. Data Science Made Simpler: https://datasciencemadesimpler.wordpress.com/category/recommendations/ [accessed 15 April 2019]

Ricci, F., 2015. Part 13: Item-to-Item Collaborative Filtering and Matrix Factorization. retrieved Feb, 13.

Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S. (2007) Collaborative Filtering Recommender Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) *The Adaptive Web. Lecture Notes in Computer Science, vol 4321*. Springer, Berlin, Heidelberg

Tunalı, V., & Bilgin, T. T. (2012). Türkçe metinlerin kümelenmesinde farklı kök bulma yöntemlerinin etkisinin araştırılması. *Elektrik, Elektronik ve Bilgisayar Mühendisliği Sempozyumu(LECO 2012).*

Uluyağmur, M. (2012). Hibrit Film Öneri Sistemi (*Doctoral dissertation*, Bilişim Enstitüsü).

Vardhan, 2019, What is Selenium? Getting started with Selenium Automation Testing [online]. Available at: https://www.edureka.co/blog/what-is-selenium/ [accessed 20 April 2019]

CURRICULUM VITAE

Name & Surname	**:**	Elif Güner
Place and Year of Birth	**:**	Zonguldak 1984
Foreign Language	**:**	English
Undergraduate	**:**	Yıldız Technical University
		Electronics and Communication Engineering 2006
Working Life	**:**	Netaş – Software Engineer 10.2006 - 10.2017
		Banksoft – Software Engineer 11.2017 - Present