A Movie Recommendation System Design Using Association Rules Mining and Classification Techniques

ZAKARIA SULIMAN ZUBI[1], ALI A. ELROWAYATI[2], IBRAHIM SAAD ABU FANAS[3]
[1]Department of Computer Science, Faculty of Science, Sirte University, LIBYA
[2] Department of Electronic, College of Industrial Technology, Misurata, LIBYA
[3]Department of Information Technology, Libyan Academy, Misurata, LIBYA

Abstract: - The importance of recommendation systems is increasing day by day due to the massive number of data and information-overloaded arising from the internet. This data can be collected in predictive datasets; these datasets can be processed and analysed via data mining methods. In this paper, an efficient hybrid movie recommender system has been designed using the association rules mining technique and K-nearest neighbours (KNN) algorithm as a classification method. The K-nearest neighbours (KNN) algorithm subsystem was used to create the first candidate list through a practical MovieLens dataset, which was retrieved from the source of the NetFlix network. Besides, the Apriori algorithm subsystem is used to analyse the same dataset and create the second list. Finally, the proposed system creates a final recommended list by matching the two lists. The results of the proposed system provide better performance than the existing systems in terms of the important degree. The important degree gives a better accuracy rate than the existing techniques used.

Key-Words: -Recommendation engine, Association Rules Mining, Collaborative Filtering, Apriori algorithm, Classification.

Received: August 19, 2021. Revised: April 15, 2022. Accepted: May 12, 2022. Published: June 6, 2022.

1 Introduction

Dataset is an important factor these days, especially for many applications worldwide for many purposes such as scientific, industrial and commercial enterprises, whereas: databases stores extremely large amount of data. But this tremendous amount of data is useless and needs to be analyzed to find the hidden data to help the decision makers to come up with important decisions. In this case we need a powerful approach to analyze this data this approach is called data mining. Data mining is a method of analyzing and generating data and rules gathering, it is specialized also in identification of the relevant elements with each other. Association rule mining is one of the poplar data mining techniques that focus on finding frequent patterns, correlations, associations, or causal structures from data sets found in various types of data sets.

Movie recommendation systems provide a mechanism to assist users in classifying customers with similar interests. In [1] authors used a new approach that can solve sparsity problem to a great extent. In [2], authors built a recommendation engine by analyzing rating data sets collected from Twitter to recommend movies to specific user using R. In Golbeck and Hendler [3], they also proposed FilmTrust, which is the website that integrates Semantic Web-based social networks and augmented with trust, to create predictive movie recommendations. It works by applying a collaborative filtering where the recommendations were generated to suggest how much a given user may be interested in a movie that the user already found. In [4][5][6] authors built a movie recommender system using the K-means clustering and K-nearest neighbor (KNN) algorithms.

Recently, in [7] authors present a complete survey of recommendation systems and give a platform for researchers in the recommendation system domain and provide collective discussions over various techniques.

In this paper, we will use Apriori algorithm which is an influential algorithm for mining association rules. Meanwhile, association rules mining plays an essential role in rule-based recommendation system.

However, the classic Apriori algorithm has many advantages and disadvantages. The main downside is the degree of importance does no measured by the minimum support and confidence. Furthermore, the

Apriori algorithm deals only with single Boolean association rules [14]. However, the NetFlix database contains many characteristics and is considered multi-dimensional association rules, not single Boolean association rules.

Therefore, this paper proposes a solution to these problems by using the KNN classification algorithm with the Apriori algorithm to increase the accuracy of the recommender system. On the other hand, it increases the efficiency of the Apriori algorithm in two stages. First, the contents of the subsets are arranged. Second, the ineffective elements are removed which leads to a decrease in the efficiency of the system.

2. Recommender Systems

Recommender systems are employed to help users to find their items based on their preferences. They produce individualized recommendations as output or have the effect of guiding the user in a personalized way to find interesting or useful items in a large amount of other items [12].

To produce recommendations, these systems need background data, input data and an algorithm. Background data is the information that the system has before it produces any recommendation. Input data is the information that is communicated to the system by the user in order to produce recommendations. An algorithm in the system is needed to combine the input data and the background data to produce a recommendation. Based on these three points, mentioned by Burke in [12] it distinguished five different recommendation methods as follows:

(1) A collaborative recommender system collects ratings of items, recognizes similarities between users based on their ratings, and produces new recommendations based on inter-user comparisons.
(2) Content-based recommender systems produce recommendation based on the associated features of an item: it recognized a user's interests profile based on the features present in items that the user has rated before.
(3) A recommender system based on demographic categorizes users based on personal attributes and finds interesting items based on demographic classes.
(4) Utility-based systems evaluate the match between a user's need and the set of options available: it recommends items based on a computation of the utility of each item for the user.
(5) Knowledge-based recommenders also make such evaluations, but they have knowledge about how a particular item meets a particular user's need.

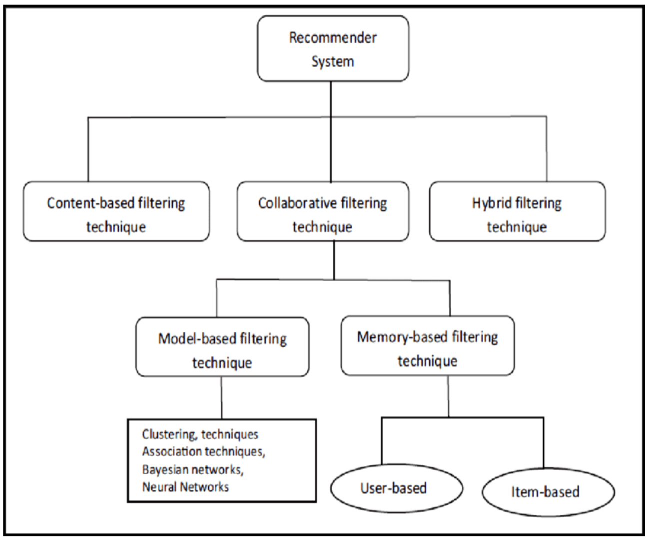

Figure 1: The main types of recommendation system

A hybrid recommender systems are developed to build a recommender system that combine two or more recommendation methods into one recommender system for a better performance. The following combination methods are identified by Burke in [1]:

(1) A weighted hybrid recommender system calculates the score of a recommended item from the results of the recommendation methods that the system uses.
(2) Switching hybrid recommender systems uses some criterion to switch between the recommendations methods used in the system to do the recommendation
(3) In a mixed hybrid recommender, recommendations from the different recommendation methods are presented together.
(4) Hybrid recommender systems based on feature combination combine the features of the unlike recommendation methods in the system and use these features in a single

recommendation algorithm to produce recommendations.

(5) In a cascade hybrid recommender system, one recommendation method is used first to produce a ranking of recommended items and a second recommendation method refines this ranking of items.

(6) A hybrid recommender based on feature augmentation method uses the output of one recommendation method as input for another recommendation method used in the recommender system.

(7) Meta-level hybrid recommenders use the model learned by the first recommendation model as input to another recommendation method.

The proposed hybrid movie recommenders [13] also combined the content-based method with collaborative filtering to get a higher accuracy of performance. Both methods were based on a naïve Bayesian classifier and the evaluation of the recommenders, it combined the movie data from IMDb as well as the rating data from Netflix. In Symeonidis et al. [13], they constructed a feature-weighted user profile to disclose the duality between users and features. The outline of their approach consisted in four steps:

(1) Constructing a content-based user profile from both collaborative and content features;

(2) Quantifying the affect of each feature inside the user's profile and among the users;

(3) Creating the user's neighborhood by calculating the similarity between each user to provide recommendations;

(4) Providing a Top-N recommendation list for each test user based on the most frequent feature in his neighborhood. The experimental results were performed with IMDb and MovieLens data sets.

3 Association Rule Mining

In general, association rule mining is the process of finding association rules. An association rule is an expression on the form $X \Rightarrow Y$. This rule can be read as: "IF X THEN Y", where X and Y are sets of items in the database. With such rule there are measures of worthiness associated with it. These measures are being support s and confidence c. The calculation of the support(s) and confidence(c) is performed as follows (1) (2) (3):

$$\text{Confidence}\ (X \Rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)} \tag{1}$$

$$\text{Suppot(x)} = \frac{Number\_of\_transectio\_contain\_item\_X}{Total\_number\_of\_transaction\_in\_dataset} \tag{2}$$

$$\text{Suppot(x Uy)} = \frac{Number\_of\_transectio\_contain\_item\_X\_and\_Y}{Total\_number\_of\_transaction\_in\_dataset} \tag{3}$$

For example, suppose that we would like to determine which items are frequently purchased together within the same transactions in a computer firm and suppose that we have found the following rule:

Contains (T, "computer")⇒Contains (T, "software")
[Support = 1%; confidence = 50%]

The interpretation of such rule is as follows: 50% of transactions, T, which contains computer, also contain software. 1% of all transactions, T, contain both of these items. In our work we will use association rules mining as an essential role in our proposed rule-based recommendation system. The association rules will be generated by a common known algorithm for association rules mining called Apriori algorithm.

4 Apriori Algorithms

The Apriori algorithm was proposed by Agarwal and Srikant in 1994. Apriori is intended to operate on databases or datasets containing transactions, Apriori in[12], is an algorithm for frequent item set mining and association rule learning over transactional databases. It profits by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

The algorithm is known also as a classic algorithm for learning association rules. Besides, the Apriori algorithm is applied on a database that contains the transaction (e.g. a collection of items purchased by customers etc.). It is also easy to execute and very simple. It is used to mine all frequent item sets in database. The algorithm makes many searches in database to find frequent item sets whereas; k-item sets are used to generate k+1-itemsets. Each k-item set must be greater than or equal to minimum support threshold frequency. Otherwise, it is called candidate item sets. In our proposal work we will use Apriori algorithm in generating association rules to find frequency of 1-itemsets that contains only one item by counting each item in the MovieLens dataset. The frequency of 1-itemsets is used to find the item sets in 2-itemsets which in turn is used to find 3-itemsets and so on until there are not any more k-item sets. If an item set is not frequent, any large subset from it is also non-frequent. In this condition pruning from the search space in MovieLens dataset is conducted. Figure 2, illustrates the flowchart of Apriori Algorithm [10], [11].

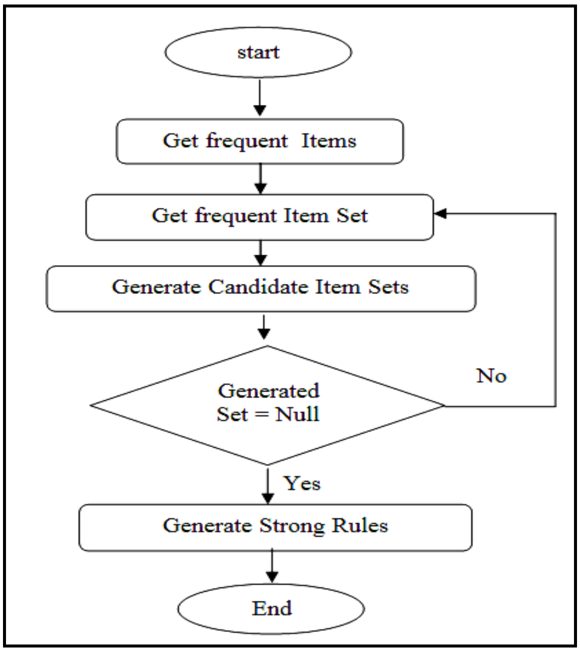

Figure 2: Flowchart of Apriori algorithm

5. K-Nearest Neighbours Algorithm

The k-nearest neighbours (KNN) algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

For classification problems, a class label is assigned on the basis of a majority vote for instance, the label that is most frequently represented around a given data point is used. While this is technically considered “plurality voting”, the term, “majority vote” is more commonly used in literature. The distinction between these terminologies is that “majority voting” technically requires a majority of greater than 50%, which primarily works when there are only two categories. When you have multiple classes for example, four categories, you don’t necessarily need 50% of the vote to make a conclusion about a class; you could assign a class label with a vote of greater than 25%.

5.3 Compute KNN Using Distance Metrics

The main goal of the k-nearest neighbour (KNN) algorithm is to identify the nearest neighbours of a given query point, so that we can assign a class label to that point. In order to do this, KNN has a few requirements these requirements are indicated as following:

1. Determine your distance metrics

In order to determine which data points are closest to a given query point, the distance between the query point and the other data points will need to be calculated. These distance metrics help to form decision boundaries, which partitions query points into different regions. A commonly decision boundaries will be visualized with a Voronoi diagrams.

2. Euclidean distance

This is the most commonly used distance measure, and it is limited to real-valued vectors. Using the below formula (4), it measures a straight line between the query point and the other point being measured.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \qquad (4)$$

3. Compute KNN Defining K

The k value in the k-NN algorithm defines how many neighbours will be checked to determine the classification of a specific query point. For example, if k=1, the instance will be assigned to the same class as its single nearest neighbour. Defining k can be a balancing act as different values can lead to over fitting or under fitting. Lower values of k can have high variance, but low bias, and larger values of k may lead to high bias and lower variance.

The choice of k will largely depend on the input data as data with more outliers or noise will likely perform better with higher values of k. Overall, it is recommended to have an odd number for k to avoid ties in classification, and cross-validation tactics can help you choose the optimal k for your dataset.

6 The Proposed Movie Recommendation System

In this section, the parts of the proposed system will be explained in Figure (3) and it combines two different techniques; collaborative filter and and association rules. The collaborative filter based on calculating the similarity between films and the characteristics of the movie type, average rating. Meanwhile, the association rules according to support and confidence using the Apriori algorithm will be me measured as well.

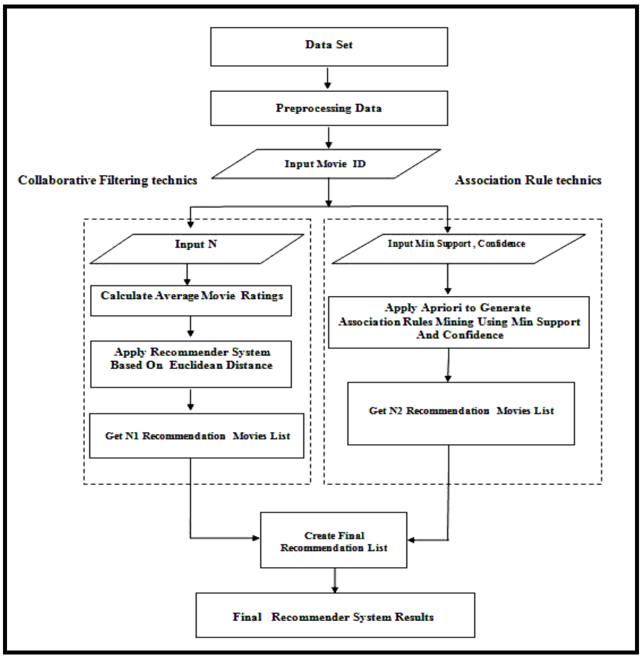

Figure 3: Proposed Movie Recommendation System

Dataset

We used in this paper, Netflix Movies dataset and it contains data of users who watch movies and detailed movie data, in addition to 100,000 records of movie viewers form 943 users and 1682 movies.

Preprocessing Data

In order to increase the efficiency of the Apriori algorithm preprocess stage applied on dataset in two sub-stages. First, the contents of the subsets are arranged. Second, the ineffective elements are removed. It presence leads to decrease the efficiency of the system.

In sorting sub-stage, the data will be sorted in ascending order and grouped according to the sequence of users. In removed redundancy sub-stage, for each user, remove the ineffective elements due to slow in execution of the classic Apriori algorithm because it always scans the elements every time in all dataset, the unsorted elements were consuming time and effort in the implementation [14]. Therefore, the proposed system sorts the elements, and removing the elements that do not affect the results.

Input Movie ID

In this stage, the user selects the movie id in order to calculate the similarity with the selected movie, and the rest of the movies in the recommendation system, whether in the part related to the rules of association or in the part about calculating similarity in the recommendation system.

The collaborative filter subsystem consists of four components as follow:

First, input the number of Movies N in candidate list:

At this stage, we input the number of Movies list that the system will propose after applying the recommendation system.

Second, calculate Average Movie Rating:

In the recommendation system, the average rating of the Movie will be calculated based on the rating by other users; the calculation will be placed in the dataset. The data will be grouped by the movie ID, to compute the total number of ratings (each movie's popularity) and the average rating for every movie. On the other hand, we will determine a list of users similar to a user U that we need to calculate the rating R. Whereas; the user U would give to a certain item I. Again, we will repeat this procedure many times just like similarity; you can do this in multiple ways.

We can predict that a user's rating R for an item I will be close to the average of the ratings given to I by the top rating 5 or top rating 10 users most similar to U. The mathematical formula for the average rating given by n users are indicated as the following:

$$R_u = \left(\sum_{u=1}^{n} R_u\right)/n \tag{5}$$

This equation (5) shows that the average rating given by the n similar users is equal to the sum of the ratings given by them divided by the number of similar users, which is n. There will be situations where the n similar users that you found are not equally similar to the target user U. The top rating 3 of them might be very similar, and the rest might not be as similar to U as the top rating 3. In that case, we could consider an approach where the rating of the most similar user matters more than the second most similar user and so on. The weighted average can help us achieve that[12].

Third, Apply Recommendation System Based on Euclidean Distance

Applying a recommendation system based on the Euclidean distance algorithm to calculate the similarity between movies related to the user's desire, using the characteristics of the movies types (action, Documentary, Romance, ... etc.). The Euclidean distance is a familiar distance measures used for 2- dimensional and 3-dimensional geometry. The Euclidean distance r2(x, y) between two 2-dimensional vectors x = (x1, x2)T and y = (y1, y2)T is given by the following equation:

$$r2(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \tag{6}$$

Define a function that computes the "distance" between two movies based on how similar their genres are, and how similar their popularity is. Just to make sure it works, we'll compute the distance between movies ID in the next step.

Forth, Generate N1 recommendation movies list

The recommendation system determines the movies that are most similar to the user's request and puts them in a list called N1.

In the second part, the association rules subsystem consists of four components as follow:

<u>Part one</u>; select support and confidence terms: select them is one of the necessary of Apriori algorithm.

<u>Part two</u>; apply Apriori to generate association rules mining using min support and confidence

The Apriori algorithm used to create association rules, between movies, according to the support and trust specified by the user.

<u>Part three</u>; generate The N2 recommendation movies list

Define the list of movies to appear based on the association rules called N2 list.

<u>Part four</u>; create final recommendation list

Match the two Movie lists; N1 of the collaborative recommender system and the list N2 based on Apriori algorithm in order to create fully recommended list, the final list is the proposed results of proposed recommender system.

7 Implementation

The proposed system was implemented using the C# programming language, to demonstrate the

Apriori algorithm, illustrated in Figure (5), where the user can specify the required support and confidence, and the system finds association rules between the elements in the Netflix dataset.

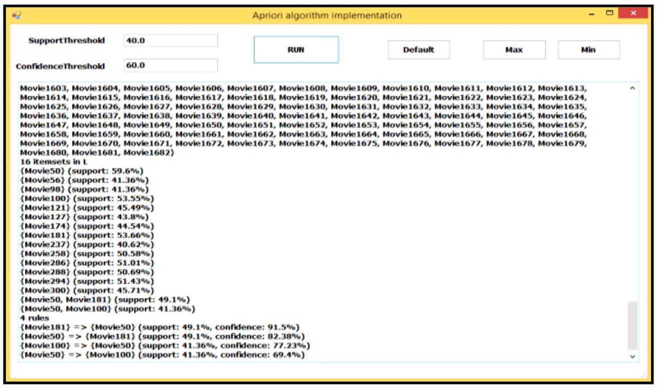

Figure 5: Apriori algorithm implementation.

Support and confidence values can be modified as desired, and the system finds association rules between items, which is a recommendation system, based on association rules, using the Apriori algorithm. The python language was used as well to implement the part of the recommendation program based on the KNN algorithm using the Euclidean distance measures Since the python language contains many libraries for machine learning systems, which greatly helped in completing the recommendation system and achieving a promising recommendations result.

8 Results and Discussion

This section presents an experimental study of our proposed system. It presents the experiment results, and summarizes our observation. The performance of our system evaluates the dataset based on the important degree term.

There are some cases that have been observed to validate our system. First, we apply the Apriori algorithm for several different users on the Netflix dataset with minimum support was 50, and a confidence value 60%. According to that we obtained a list, containing most ranked of the movies, that users interacted with in the dataset of the Netflix as shown in table 1. We observed from table 1, the most frequently movies by users. It also could be notable that the capability of Apriori algorithm of extract the movies have frequently used in dataset. However, this list has a minimum number of items up to 21 movies only among those in the dataset. Thus, we have obtained the nearest neighbour movies to support this list.

Movie ID	Movie Title
1	Toy Story (1995)
7	Twelve Monkeys (1995)
15	Mr. Holland's Opus (1995)
50	Star Wars (1977)
56	Pulp Fiction (1994)
64	Shawshank Redemption
89	Blade Runner (1982)
96	Terminator 2: Judgment Day (1991)
98	Silence of the Lambs
121	Independence Day (ID4) (1996)
172	Empire Strikes Back
173	Princess Bride
174	Raiders of the Lost Ark (1981)
181	Return of the Jedi (1983)
222	Star Trek: First Contact (1996)
227	Star Trek VI: The Undiscovered Country (1991)
228	Star Trek: The Wrath of Khan (1982)
229	Star Trek III: The Search for Spock (1984)
230	Star Trek IV: The Voyage Home (1986)
258	Contact (1997)

Table 1, the most ranked videos in the Apriori algorithm

Second, when applying the KNN algorithm, and assuming that the value of k =15. If we choose a movie entitled "Star Wars (1977) " movie as an example. The movie video ID = 50, we have a list of recommended movies from the system, shown in the table 2. This list is the recommended movies using KNN to the movie entitled " Star Wars (1977)".

Return of the Jedi (1983)	4.0
Empire Strikes Back, The (1980)	4.2
Starship Troopers (1997)	3.2
Independence Day (ID4) (1996)	3.4
African Queen, The (1951)	4.1
Star Trek: First Contact (1996)	3.6
Jurassic Park (1993)	3.7
Star Trek: The Wrath of Khan (1982)	3.8
Raiders of the Lost Ark (1981)	4.2
Star Trek IV: The Voyage Home (1986)	3.4
Star Trek III: The Search for Spock (1984)	3.1
Star Trek VI: The Undiscovered Country (1991)	3.2
Indiana Jones and the Last Crusade (1989)	3.9
English Patient, The (1996)	3.6
Princess Bride, The (1987)	4.1

Table 2, the list of recommended movies from KNN to Star Wars (1977)

On the other hand, when we use Apriori algorithm and chooses the movie ID =50 entitled "Star Wars (1977)", we will found the 10 most related movies based on the Apriori algorithm. table 3 shows the matched movies according to Apriori algorithm and KNN.

Movie Title	Rating
Return of the Jedi (1983)	4.0
Empire Strikes Back, The (1980)	4.2
Independence Day (ID4) (1996)	3.4
Star Trek: First Contact (1996)	3.6
Star Trek: The Wrath of Khan (1982)	3.8
Raiders of the Lost Ark (1981)	4.2
Star Trek IV: The Voyage Home (1986)	3.4
Star Trek III: The Search for Spock (1984)	3.1
Star Trek VI: The Undiscovered Country (1991)	3.2
Princess Bride, The (1987)	4.1

Table 3, the matched movies list based on the movie entitled " Star Wars (1977)" using Apriori algorithm and KNN

The matching ratio was (10/15) = 0.666 when comparing the list of Apriori algorithm to KNN lists. This result is considered as an excellent match or high important degree as shown in Figure 6.

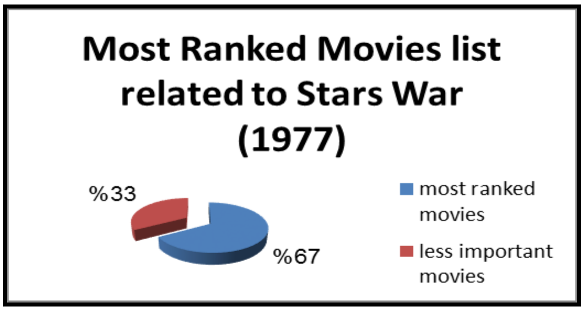

Figure 6, Most Ranked Movies list related to Stars War (1977) using Apriori algorithm.

Moreover, if we take movies ID=222 entitled "Star Trek: First Contact (1996)" as an example, the recommended KNN list by using our proposed system, the list will be as follows in table 4.

Movie Title	Rating
Jurassic Park (1993)	3.7
Star Trek: The Wrath of Khan (1982)	3.8
Star Trek IV: The Voyage Home (1986)	3.4
Star Trek III: The Search for Spock (1984)	3.1
Star Trek VI: The Undiscovered Country (1991)	3.2
Stargate (1994)	3.1
Star Trek: The Motion Picture (1979)	3.0
Star Trek: Generations (1994)	3.3
Star Trek V: The Final Frontier (1989)	2.3
Judge Dredd (1995)	2.8
Time Tracers (1995)	1.5
Indiana Jones and the Last Crusade (1989)	3.9
Raiders of the Lost Ark (1981)	4.2
Men in Black (1997)	3.7
Starship Troopers (1997)	3.2

Table 4, the list of recommended movies from KNN to Star Trek: First Contact (1996)

Once we compare these results with cluster list in KNN algorithm for movie ID=222 we found 5 movies only related to the move entitled " Star Trek: First Contact (1996)" which is the most ranked movies Figure 7 shows the most ranking movies of the mentioned movie. The list is shown in table 5.

Movie Title	Rating
Star Trek: The Wrath of Khan (1982)	3.8
Star Trek IV: The Voyage Home (1986)	3.4
Star Trek III: The Search for Spock (1984)	3.1
Star Trek VI:The Undiscovered Country (1991)	3.2
Raiders of the Lost Ark (1981)	4.2

Table 5, The matched movies list to Star Trek: First Contact (1996)

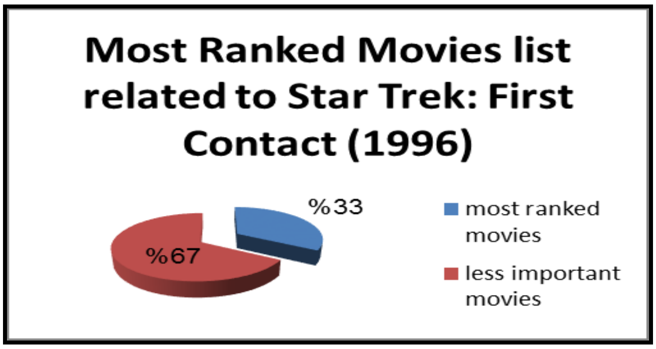

Figure 7, The most ranked movies list related to the movie entitled " Star Trek: First Contact (1996)"

This matching ratio was (5/15) = 0.333, which is considered as a good match between two Lists. Therefore, when we applied KNN to the movie ID= 258 entitled "Contact (1997)", the outcome list of movies will be shortlisted as follows in table 6.

Movie Title	Rating

Twelve Monkeys (1995)	3.7
Day the Earth Stood Still, The (1951)	3.9
Until the End of the World (Bis ans Ende der Welt) (1991)	2.8
Dead Man Walking (1995)	3.8
Mr. Holland's Opus (1995)	3.7
Shawshank Redemption, The (1994)	4.4
One Flew Over the Cuckoo's Nest (1975)	4.2
Dead Poets Society (1989)	3.9
Trainspotting (1996)	3.8
Time to Kill, A (1996)	3.6
It's a Wonderful Life (1946)	4.1
Clockwork Orange, A (1971)	3.9
To Kill a Mockingbird (1962)	4.2
People vs. Larry Flynt, The (1996)	3.5
Field of Dreams (1989)	3.6

Table 6, The recommended movies list using KNN to movie " Contact (1997)"

When we compare the inferred used by the KNN algorithm for movie ID=258 entitled " Contact (1997) " we will found 2 movies related to that movie. This proves that the comparison has been done more accurately. Those movies are shown in table 7 and as well as the important degree ratio in are illustrated in Figure 8.

Movie Title	Rating
Twelve Monkeys (1995)	3.7
Shawshank Redemption, The (1994)	4.4

Table 7, The matching movies list of the movie entitled " Contact (1997)"

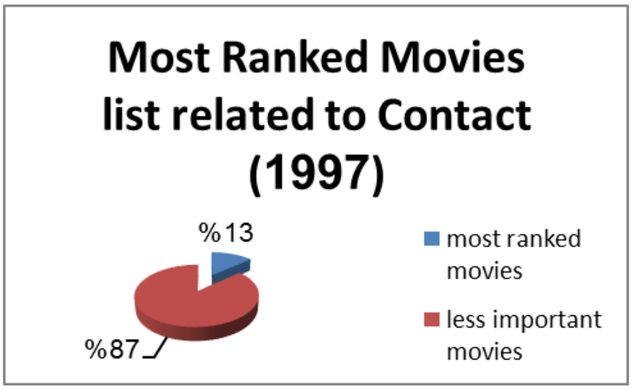

Figure 8, The most ranked movies list related to the movie entitled "Contact (1997)"

Based on that the matching ratio was (2/15) = 0.133, which is considered too bad matching between two lists.

Therefore, the number of most ranked videos by using Apriori algorithm list is very small due to the disadvantage of using Apriori algorithm alone in term of the important degree. At our discretion, we suggest supporting the most ranked list of Apriori algorithm by adding the related videos found in the KNN list with this we give the user more recommended videos based on his first chosen movie.

Thus we can conclude, that the new movie which has less number of using, the Apriori algorithm cannot meets the minimum supportive degree accurately. Therefore, the Apriori algorithm list could be supported by nearest neighbour movies extracted by KNN technique. As an example, movie entitled "Contact (1997)" has minimum number of related movies in Apriori list as it is shown in table 7. Therefore, we can support this list by adding the nearest neighbor items in KNN list in table 6.

9 CONCLUSIONS

In this paper, an efficient hybrid movie recommender system has been designed using the association rules mining technique and collaborative filter technique. The data were taken from Movielens dataset and the system were implemented in the Python and C# programming languages. A dataset was taken from the MovieLens dataset granted from Netflix.Our proposed recommendation system applied the KNN algorithm as a classification method as well as the Apriori algorithm as an association rules mining. Applying both techniques give more realistic movie lists for the user to choose. The results were evaluated in term of the important degree. The proposed system improves the important degree and gives better accuracy than the existing techniques used. KNN and Apriori algorithm improved the lists of user-recommended movies that are close to their liking, depending on which movie the user selects the first time. In the future, the proposed system can be more improved using big datasets. In addition, new directions for improvement could be using deep learning techniques which may enhance the efficiency of the movie recommendation system, in that case the model can be tuned to trained more situations.

ACKNOWLEDGEMENTS

The authors would like to thank, the Department of Computer Science at Faculty of Science, Sirte

University, Libya, College of Industrial Technology, Misurata, Libya, and The Libyan Academy Department of Information Technology, Libya. Furthermore, Full thanks to the Ministry of Higher Education, Libya for partially supported financial support.

References:

[1] Mishra N., Chaturvedi S., Mishra V., Srivastava R., Bargah P. (2017)Solving Sparsity Problem in Rating-Based Movie RecommendationSystem. In: Behera H., Mohapatra D. (eds) Computational Intelligence in Data Mining. Advances in Intelligent Systems and Computing, vol 556. Springer, Singapore

[2] Das D., Chidananda H.T., Sahoo L. (2018) Personalized Movie Recommendation System Using Twitter Data. In: Pattnaik P., Rautaray S., Das H., Nayak J. (eds) *Progress in Computing, Analytics and Networking.Advances in Intelligent Systems and Computing, vol 710*. Springer,Singapore

[3] Golberg, J., Hendler, J. (2006). FilmTrust: movie recommendations using trust in web-based social networks. *In Consumer Communications and Networking Conference, Vol. 1, (pp. 282-282).*

[4] Ahuja, R., A. Solanki, and A. Nayyar. Movie recommender system using k-means clustering and k-nearest neighbor. *in 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). 2019. IEEE.*

[5] Kokate, S., et al. Traveler's Recommendation System Using Data Mining Techniques. in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). 2018. IEEE.

[6] Li, H. and D. Han, A Novel Time-Aware Hybrid Recommendation Scheme Combining User Feedback and Collaborative Filtering. IEEE Systems Journal, 2020.

[7] Awati, C. and S. Shirgave. The State of the Art Techniques in Recommendation Systems. in International Conference on Computing in Engineering & Technology. 2022. Springer.

[8] Ye, Y. Research on Apriori algorithm and its application in electronic commerce system. *in 2016 International Conference on Advances in Management, Arts and Humanities Science (AMAHS 2016). 2016.* Atlantis Press.

[9] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *In User Modeling and User-Adapted Interaction, 12, (pp. 331–370).*

[10] Mendes, R. I. (2007). "A Hybrid Recommender for movies based on Naïve Bayesian Classifier." *Bacherlor's Thesis Informatics & Economics 2007, Erasmus University Rotterdam*.

[11] Symeonidis, P., Nanopoulos, A., Manopoulos, Y. (2007). Feature-Weighted User Model for Recommender Systems. *In Proceedings of the 11th International Conference on User Modeling, (pp. 97-106).*

[12] Rakesh Agrawal and Ramakrishnan Srikant Fast algorithms for mining association rules in large databases. *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499,* Santiago, Chile, September 1994.

[13] Karandeep, T., Abhishek N and Mahajan Narsale. ,Recommendation System using Apriori Algorithm. IJSRD - *International Journal for Scientific Research & Development*

| Vol. 3, Issue 01, 2015 | ISSN (online): 2321-0613.

[14] Zakaria Suliman Zubi, Ayman Altaher Mahmmud, Crime Data Analysis Using Data Mining Techniques to Improve Crimes Prevention, *international journal of computers, ISSN: 1998-4308, Volume 8, 2014*.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Zakaria Suliman Zubi, carried out the optimization as well as the statistics of the article.

Ali A. Elrowayati, carried out the evaluation of the system performance as well as prepared the statistics of the article results.

Ibrahim Saad Abu Fanas carried out the idea and implemented the algorithm's code with Python and C# programming language.

Sources of funding for research presented in a scientific article or scientific article itself

The research work was partially supported by the Ministry of Higher Education, Libya.