

# Implementing Recommender Systems using Machine Learning and Knowledge Discovery Tools

Mohammad Zahrawi<sup>1,\*</sup>, Ahmad Mohammad<sup>1</sup>

<sup>1</sup> Faculty of Engineering & IT, British University in Dubai, Dubai, UAE; 20209704@student.buid.ac.ae

\* Correspondence: 20000050@student.buid.ac.ae

Manuscript received: 02-08-2021, revised: 27-08-2021, accepted: 28-08-2021.

## Abstract

The current research offers a novel use of machine learning strategies to create a recommendation system. At recent era, recommender systems (RSs) have been used widely in e-commerce, entertainment purposes, and search engines. In more general, RSs are set of algorithms designed to recommend relevant items to users (movies to watch, books to read, products to buy, songs to listen, and others). This article discovers the different characteristics and features of many approaches used for recommendation systems in order to filter and prioritize the relevant information and work as a compass for searching. Recommender engines are crucial in some companies as they can create a big amount of income when they are effective or be a way to stand out remarkably from other rivals. As a proof of the importance of recommender engine, it can be stated that Netflix arrange a challenge (the “Netflix prize”) where the mission was to create a recommender engine that achieves better than its own algorithm with a prize of 1 million dollars to win.

**Keywords:** Collaborative filtering; content based; memory-based; hybrid recommender system; item-based; user-based.

## 1. Introduction

Natural language processing (NLP) is a core domain for several applications [1], [2]. One of such applications is the recommender systems [3]. The task of a RS is to create useful recommendations to a group of users for products or items that might interest them. Suggestions a book on Amazon maybe one of the major interests for readers, or films on Netflix might be audience would like it, are real life world examples of the RSs . The availability of data in certain domain used in design a recommendation engine [4]. For instance, users on Netflix frequently rate movies on a scale of 1 (bad) to 5 (awesome) [5]. Such recording rates of movies by every user and store them in a big table [6]. Recommender engines are helpful to both users and service supplier. They help dramatically to decrease transaction costs of selecting and finding products in an online shopping website. Recommendation engines as well showed a huge improvement in decision making procedure and quality. The e-commerce’ revenues has been increased by using recommender engines, as matter of fact, they are active means of selling more items. In scientific libraries, recommender engines aid readers by letting them to move farther on catalog searches. So, the need to use effective and correct recommendation engines within an organization that will provide related and reliable recommendations for users might not be overemphasized [7].

One of the main differences between RS and search engine (information retrieval system) is user behavior, is the core of recommender engine, while it plays an essential role in search engine [8]. Search engine like Google classically focus on efficient indexing that used to find information (data) without even looking at every row in a database. Also, stemming and tokenization that utilize to match each row that related to the base form word (e.g., draw, draws, drawing, drawn), and breaking up the text into meaningful words by using tokenization process. Techniques like survey, questionnaire, feedback to empower search engine to refine its search result and work similar way to recommendation system [9]. All the recommendation algorithms have strong point and weaknesses, and many researchers suggest combine these algorithms to achieve highest performance. For this motive, the current research was showed many recommendation methods being researched, analyzing them in aspect of data source that improve the recommendations and the algorithms that run on that data. hybrid with collaborative filtering improved the performance of our knowledge-based RS Entrée.

Recommendation engine based mainly machine learning algorithms which is subfield of Artificial Intelligence. It can scan through all possible products and create a personalized listing of items which you might be interested in. There are two widely techniques used to analyze data sources in RSs are collaborative filtering and knowledge-based approaches. They used to improve notions of affinity between items and users which can be used to recognize a perfect pair of items. however, collaborative filtering is used to handle the similarity

between users to make recommendations based only on the past histories recorded between users and items in order to create new suggestions. These histories are stored in “user-item interactions matrix” [10].

## 2. Related Work

Machine learning models recently have grown dramatically due to the increase of available data that need to extract insights and knowledge from these data [11]–[15]. The applications of machine learning models have been noticed remarkably advanced in multiple engineering applications [16]–[20]. In this article, machine learning model is used to show the set of analysis method that utilized to deduce recommendation techniques or build recommendation engine from huge datasets [21]. Recommender engines are most known intelligent software systems that are employed in many fields such as music, movies, restaurant, financial services, and Twitter followers [22]. It suggests interesting items to users. These recommendations are valuable alternative to searching engine techniques. Richard [23] defined a personalized recommender engine to customers in supermarkets. This recommender engine depends on customer previous history towards the purchases to recommend new items for them. The IBM investigators established this recommender engine to implement it as a portion of SmartPad which was advanced as a personal helper for online shopping.

In another study, the authors showed a hybrid book recommender engine built on author's rankings by users and Collaborative Filtering [24]. This hybrid recommender engines enhance book suggestion through book's reader received proposals from hybrid engine to choose which book to read next. MovieLens dataset is the online website run by GroupLens Research at the University of Minnesota. MovieLens suggests some of the most popular movies to new users to evaluate it. regard to your movie ratings, MovieLens creates personalized forecasts for movies you might enjoy, as well help you to avoid movies that you won't. These ratings are utilized to suggest other movies to the user. In addition, MovieLens uses Collaborative Filtering based on these ratings to create personalized recommendations.

Frequently, users don't have any idea about what products are useful (good) for them, which of these products are not. However, the racing between companies makes recommender engines of special interest. These companies contest with each other's to increase their sales and satisfy consumers by providing them expectable (suitable) items in their life needs. Moreover, these companies do not know what is suitable to consumers. So, it is significant these companies use the best recommender engines to display the exact recommendations for users to rise their sales and therefore increasing their revenue [25].

### I. Content Based

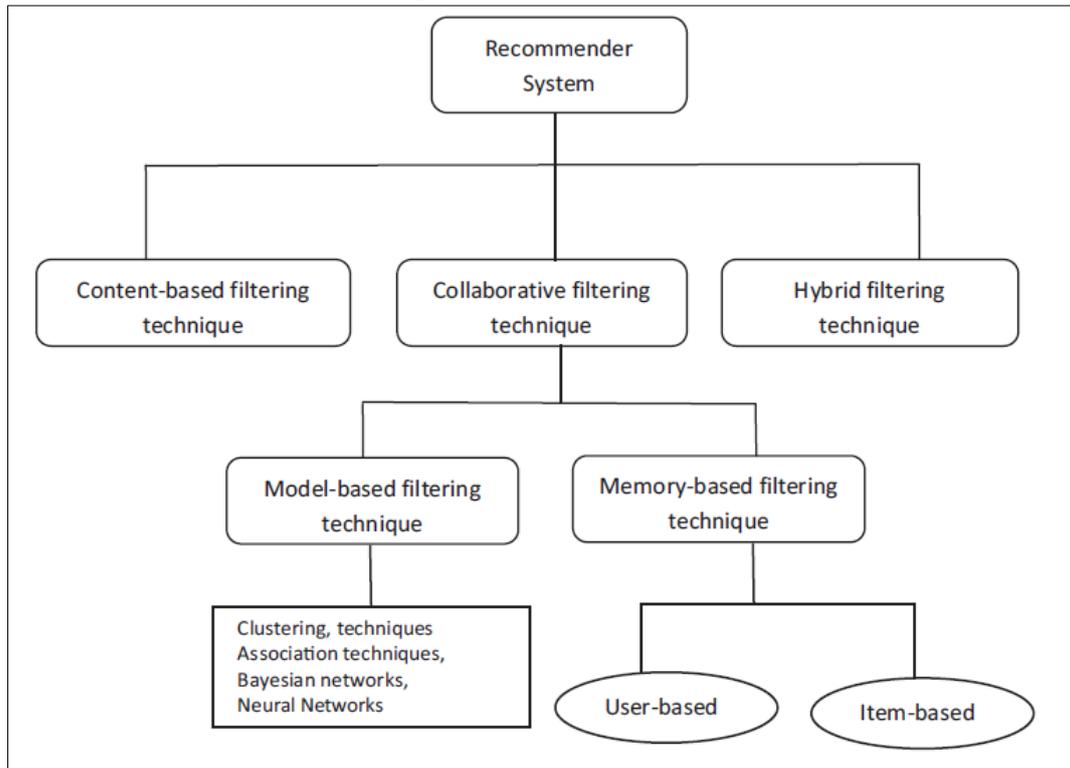
In contrast Collaborative Filtering techniques that only manipulate the user ratings matrix. Content based method treat all products and items as ordinary units, where predictions are made without looking at specifics of each product or users. However, this method tries to create a model based on user's features or even information about items. These features could be age, Sex, Job, nationality, genre of a movie, the director, or any other personal features about users/products [26]. For example, given game genre information, and knowing that a player liked “call of duty game” and “battlefield game”, one may infer that the player prefers playing actions games, and could hence recommend “Halo game”. Content-based methods provide suggestion by comparing the description of product to other description of that interests the user [27].

More research has been conducted in on recommending items within regards of textual content, like website pages, newspapers, and books; where the internet sites themselves or associated content like user reviews and descriptions are available. Some methods have labelled this method as an Information Retrieval (IR) task, where the information related to user's preferences is considered as a query, and the unrated items are linked to the similarity to this query [28]. In NewsWeeder [29], each document is classified into many categories based on rating, then transformed to Tf-Idf word vectors, and computed the averaged of vectors for each category. To classify a new item based on content, it is compared with each averaged vector and assign rating according to cosine similarity for each category.

Another way to treat recommendations is employing classification algorithms, where items descriptions represent the data, and a user's ratings are considered as labels for these data. In the scope of book recommending, Mooney et al. used textual content such as author, the title, reviews, synopses, and subject terms, to build a multinomial naïve Bayes model. Ratings varies from scale of 1 to k which can be straightly mapped to k labels, otherwise these ratings could be utilized to find the weight of training data in a probabilistic binary classification setting [30]. Other classification algorithms have as well been exploited for content-based recommender engines, decision trees, including k-nearest neighbor, and neural networks [31].

### II. Collaborative filtering

The goal of this approach is offering consumers products they have not yet seen it, based on the rating history and tastes of similar users' profiles, the similarity between different consumers as well as products (items, videos, pages, books) that are used in recommended system [32]. Collaborative Filtering cannot apply semantics and systematically analyzes for products. So, Collaborative Filtering has been employed accurately and quickly in recommendations without seen any information about product(item) itself. There are two main categories for Collaborative Filtering: memory-based and model-based method (Figure 1). Where Memory-based methods utilize the directly the data that obtained from consumers to measure the affinity between consumers or products to predict the recommendations. However, memory-based techniques used only those data from consumers, which enhance the efficiency of method implementation and make it easier.



**Figure 1.** The main types of recommendation system.

Model-based technique used data to create matrix where columns represented products and also rows represented users with latent characteristics. Whereas latent factor basically a database of products and consumers usually stored the latent features. Latent variable is another name for latent factor that cannot be detected right away. Also, it is possible to used observed variables to derive latent factor. For instance, in the movie's recommender engine, the latent features may be animation, crime or documentary. Then, recommender engine can predict new movies relying on the latent factor models [10].

### 2.1 Memory based techniques

The main idea of memory-based method is that users with same traits having similar taste. For instance, if you are interested in recommending a song to our friend Jack, suppose Jack and I have seen many songs together and we rated them nearly same. So, by using this similarity, the recommendation engine would suggest new songs to Jack that I like them. So, User based collaborative filtering method measures the similarity between consumers by comparing the ratings on the same product, then forecasting the rating for a product by the active consumer would be an average of the ratings of the product by other consumers those similar to the active user. Meanwhile, Item-based filtering methods find the predictions of new items using the similarity between products. This method restores all items that rated by an active consumer from the user-item matrix, it shows how affinity the restored products are to the target product, then it picks the k most similar product and their similarities are decided as well. Prediction is done by finding a weighted average of the active consumer rating on the similar products k. Many types of similarity measures are used to calculate similarity between product/consumer. The two most known similarity measures are cosine-based and correlation-based as discussed later [7].

### 2.2 Model-based techniques

This method utilizes machine learning, machine learning algorithms to learn a model to predict consumer's rating for unrated product. These algorithms can rapidly suggest a group of products which are similar to neighborhood-based recommender methods. There are many model-based algorithms such as Clustering models, Bayesian Networks, latent semantic technique and Dimensionality Reduction technique such as Singular Value Decomposition (SVD). However, the goal of implementing dimensionality reduction techniques is to improve efficiency and accuracy of memory-based method. On this matter, algorithms like singular value decomposition, principal component analysis, described as latent factor models, which can create a low-dimension representation in terms of latent factor by compressing user-item matrix. One benefit of using this algorithm is that dealing with a low dimensional matrix that does not contain a lot of missing values, also could be applied in either user-based or item-based. Also, it could process the sparsity problem of the matrix more efficient than memory-based ones.

### 3. Challenges and Limitations

Collaborative Filtering methods have some significant privileges over content-based method in that it can proceed in domains where textual content is not abundant for each product and where textual content is difficult for a collaborative filtering to analyze such as opinions and comments. Also, CF method has capability of providing coincidental recommendations, that means it can suggest products are related the user even without textual content being available in the user's profile. Nevertheless, collaborative filtering reveals some potential problems like cold start, sparsity, scalability as described below [6]:

#### 3.1 Cold start problem.

The first issue appears in Collaborative Filtering technique, where a product cannot be suggested unless some consumer has rated it before. This problem does not appear only on new products, but also on obscure products. The profile of such new product or user will be blank since he has not rated any products; thus, his taste is unknown to the recommendation system [7]. This problem can occur under three cases:

**New user:** When a user register and create account with a service, there is no any recommendation to be generated because lack of ratings. For instance, a new consumer to Amazon has no ratings in the history, so a neighborhood of similar consumers cannot be detected. This may be fixed in various ways:

- let the consumer rate some products before first time using the service.
- showing non-personalized recommendations until the consumer rated more products.
- inquiring the consumer to describe their preferences in aggregate, e.g., "I like documentary movies".
- collect demographic information about the consumers.
- exploitation ratings of other consumers who have the same demographics to generate recommendations.

**New Item:** When a new product is added to a recommendation system, it will not be suggested to other consumers because it lacks ratings. For instance, MovieLens is not capable of recommending new movies recently release until another user rate it. Unluckily, in many fields, consumers are less probably to rate products that are not recommended to them. However, in fields with high product turnover like newspaper, the cold-start issue can be particularly irritating.

**New Community:** The major cold-start problem is bootstrapping which relate to the startup of the RS, when almost no data available the recommender can rely upon in present. This scenario shows the drawbacks of both the new user and the new item scenarios, as all products and users are new. The most popular solution for new community issue is to gather rating from small subset of community, before allowing the whole community to utilize the service.

#### 3.2 Data Sparsity problem

Data Sparsity occurs as an outcome of lack of data, which only few of rated products are available in a database [33]. This always drive to a sparse user-item matrix, which could not be able to detect successful neighbors and finally, creating weak recommendations. As a result of increasing number of consumers and products with few ratings, most data records remain zero. The level of sparsity is calculated by the dividing number of zeros in the matrix by the total number of entries.

When new products are uploaded to the system, they need to be rated by a large number of consumers before they could be suggested to consumers who have same tastes to the ones who rated them. Meanwhile, the new product problem is not impacted by content-based methods, because the recommendation of a product is based textual content rather than its ratings [9].

### 3.3 Scalability

Scalability problem arises when the time required to process large amount of data increased [34]. Because computation increased linearly with datasets contain millions of consumers and products, collaborative filtering algorithm with the complexity of (n) becomes too large. A recommendation method would be effective when it deals a limited number of datasets, but it would not be able to generate sufficient number of suggestions when the size of datasets is increased [19]. furthermore, many recommendation engines need to react instantly to online requirements and create suggestions for all consumers irrespective to their activities and ratings history, which require a higher scalability of a recommendation system. Big web companies like Twitter employ clustering algorithms to scale suggestions for their millions of users, they used huge memory machines for most computations happening in there [22].

## 4. Methodology

### 4.1 Approach

Recommendation engines typically based on large datasets and specifically need to be organized in a particular fashion. In this paper, we are going to deploy different types of RS and showing how we're going to approach creating a recommendation system with python programming language. fully developed recommendation system is actually extremely complex and resource intensive because it required heavy mathematics in linear Algebra. The two most popular kinds of recommender engines are content-based and collaborative filtering. collaborative filtering generates recommendations based on the experience of user's attitude toward products. For instance, systems in Amazon where they based on other user's shopping experiences Amazon will suggest products that it believes you will enjoy. In General, collaborative filtering (CF) is more frequently used than content-based systems because it normally gives better outcomes and it's actually relatively easy to implement from overall perspective. This paper aims to apply collaborative filtering techniques on same dataset, these techniques are:

- i. Item-Based CF.
- ii. Memory Based by computing Cosine Similarity.
- iii. Model Based by using Singular Value Decomposition.

### 4.2 Dataset

The datasets were imported from MovieLens website, which is one of the most common datasets used for creating and testing recommender engines. The datasets were collected over various periods of time, depending on the size of the set. The datasets describe ratings and free-text tagging activities from MovieLens, a movie recommendation service. First dataset contains 100,003 rows and 5 columns (user\_id, item\_id, rating, timestamp, and title), second dataset contains 1682 entries with two columns (item\_id, title). Data processing was conducting using Pandas library on Jupyter Notebook. Jupyter Notebook is an advanced integrated development environment (IDE), that enable you write python code, importing libraries that used to manipulate dataset.

### 4.3 Building a Model

- i. item-based CF

In this section, we are going to build three different type of recommender engines, let's start with basic RS item-based collaborative filtering. Table 1 reports the samples of each datasets for better understand.

**Table 1.** The samples of each datasets.

No.	User ID	Item ID	Rating	Timestamp	Title	No.	User ID	Title
0	0	50	5	881250949	Star Wars (1977)	0	1	Toy Story (1995)
1	290	50	5	880473582	Star Wars (1977)	1	2	GoldenEye (1995)
2	79	50	4	891271545	Star Wars (1977)	2	3	Four Rooms (1995)
3	2	50	5	888552084	Star Wars (1977)	3	4	Get Shortly (1995)
4	8	50	5	879362124	Star Wars (1977)	4	5	Copycat (1995)

These datasets are clean, organized, free of missing values. Table 2 presents the created user-item matrix by using pivot table function.

**Table 2.** The user-item matrix by using pivot table function.

Title User ID	'Til There was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	39 Steps, The (1935)	...
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
1	NaN	NaN	0.2	0.5	NaN	NaN	3.0	4.0	NaN	NaN	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0	NaN	...
3	NaN	NaN	NaN	NaN	2.0	NaN	NaN	NaN	NaN	NaN	...
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
6	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
...	...	...	...	...	...	...	...	...	...	...	...

As we observed from pivot table (user-item matrix), rows represent user ID and columns represent name of movies, and data here is ratings of movies, each cell will then represent of the rating the user gave to that movie. Also, Number users (rows) is 944, number of movies (columns) is 1664, that means the total number of ratings in this matrix should be 944 x 1664 or 1,570,816 if we assumed all users rated all movies. However, there are a lot of NaN values, because most people have not rated most of the movies. The percentage of NaN values around 93.6%.

Now we would use correlation between two items (movies), for example, lets find most similar movies to Starwar movie. First let's put all the user ratings for Starwar movie in an individual list (Series in python), then apply the correlations between two items (movies). However, some result does not make any sense. Because some movies are only watched once by users who as well watched Starwars. So, to fix this issue, we filtered out movies that have less than 100 reviews and the final results are reported in Table 3.

**Table 3.** The results of the filtered movies that have less than 100 reviews.

Title	Correlation	Num of rating
Star Wars (1977)	1.00	584
Empire Strikes Back, The (1980)	0.74	368
Return of the Jedi (1983)	0.67	507
Raiders of the Lost Ark (1981)	0.53	420
Austin Powers: International Man of Mystery (1997)	0.37	130

ii. Memory Based by computing Cosine Similarity

Memory-Based CF methods can be separated into two major sections: user-item filtering and item-item filtering. A user-item method focuses on “Users who are similar to you also liked X”, but item-item method focuses on “Users who liked this item also liked X”. in this section, we are going to use cosine similarity to build recommender engine. Whereas Cosine similarity is a cosine of the angle between the two vectors of the product, vectors of X and Y is calculated for imputing similarity. If the vectors are closer, then the angle would be small, and therefore the cosine would be large.

Cosine similarity is a distance metric used frequently in recommender engine, where the ratings are seen as vectors in m-dimensional space and the similarity is calculated based on the angle between these vectors. Cosine similarity for users a and u can be calculated using the formula below, where you take dot product of the user vector  $r_a$  and the user vector  $r_u$  and divide it by multiplication of the Euclidean lengths of the vectors.

$$w_{a,u} = \cos(\vec{r}_a, \vec{r}_u) = \frac{\vec{r}_a \cdot \vec{r}_u}{\|\vec{r}_a\|_2 \times \|\vec{r}_u\|_2} = \frac{\sum_{i=1}^m r_{a,i} r_{u,i}}{\sqrt{\sum_{i=1}^m r_{a,i}^2} \sqrt{\sum_{i=1}^m r_{u,i}^2}} \tag{1}$$

Finally, predictions are generally calculated as the weighted average of deviations from the neighbor's average, as in:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u \in K} w_{a,u}} \quad (2)$$

where  $p_{a,i}$  is the prediction for the active user  $a$  for item  $i$ ,  $w_{a,u}$  is the similarity between users  $a$  and  $u$ , and  $K$  is the neighborhood or set of most similar users.

### iii. Model Based by using Singular Value Decomposition

Model-based CF is based on matrix factorization (MF) which has been used widely especially as unsupervised learning techniques for dimensionality reduction and latent variable decomposition. Matrix factorization is commonly used for recommender engines where it can easily handle sparsity and scalability more than Memory-based CF. The main goal of MF is to discover the latent taste of consumers and the latent features of items from known ratings (learn features that describe the characteristics of ratings) to then predict the unknown ratings through the dot product of the latent features of consumers and product. When there are a lot of None values in a matrix, with many dimensions, it is possible to generate a matrix with low rank structure by doing matrix factorization.

For clear image about latent variables of the users and items: let's take for example the MovieLens dataset, you have the following features: (user id, location, age, gender, movie id, actor, director, language, rating, year). By employing matrix factorization on the model, we figure out that the most important user attributes are age set (under 10, 10-18, 18-30, 30-90), gender and location, and for film attributes, we discovered that actor, director and decade are most important. The interesting thing about CF that model only utilized features (user ID, movie ID, rating) to find out the latent features. If there is scarcity of data, model-based CF model will suggest badly, since it will be harder to learn the latent variables.

## 5. Singular Value Decomposition (SVD)

SVD is a matrix factorization method that is typically used to decrease the number of attributes of a dataset by minimizing space dimensions from  $N$  to  $K$  where  $K < N$ . With respect to RS, the SVD is used as a collaborative filtering method. It uses a matrix form where each row represents a consumer, and each column represents a product. The cells in this matrix are the ratings that are given to products by consumers. The factorization method is carried out by the singular value decomposition. It figures out factors of matrices from the factorization of a high-level matrices (user-item-rating). The SVD is a technique of breaking up a matrix into three other matrices as shown below:

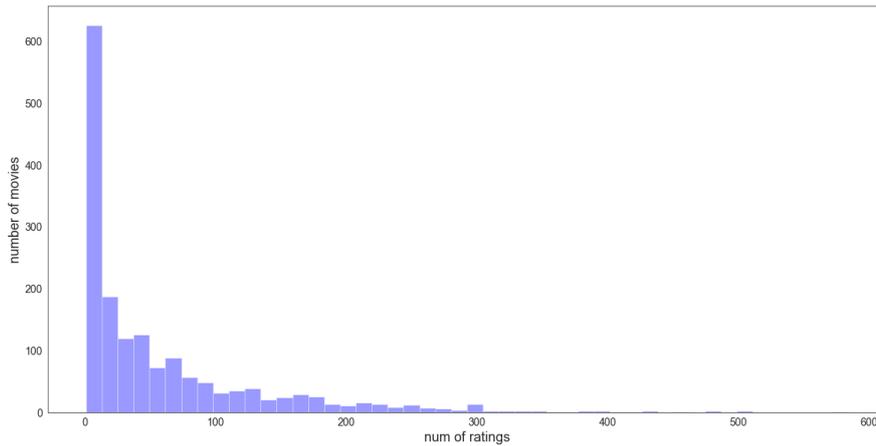
$$A = USV^T$$

- $m \times n$  is matrix  $A$ .
- $U$  is a  $m \times r$  orthogonal left singular matrix.
- $S$  is a  $r \times r$  diagonal matrix
- $V$  is a  $r \times n$  diagonal right singular matrix

Matrix  $A$  can be factorized to  $U$ ,  $S$  and  $V$ . The  $U$  matrix denotes the feature vectors corresponding to the consumers in the hidden feature space and the  $V$  matrix denotes the feature vectors corresponding to the product in the hidden feature space. SVD method may be very computationally expensive and slow. More recently, researches have been carried out to minimize the squared error by conducting alternating least square or stochastic gradient descent and utilizes regularization terms to prohibit overfitting.

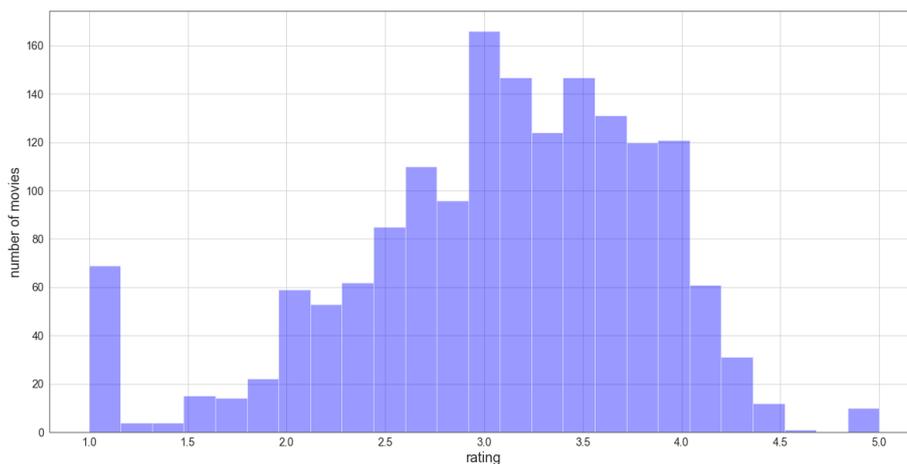
## 6. Discussion

The previous dataset contains 100,003 rows and 5 columns. By applying explanatory data analysis, like matplotlib library, seaborn library, we have observed here, the greatest number of movies have quite few of rating, 0-10 ratings, that make sense cause most people watch famous or big hit movies, those are going to be with a lot of reviews or ratings, as shown in Figure 2.



**Figure 2.** relationship between number of movies and number of ratings.

Meanwhile, figure 2 shows the distribution of ratings, you probably notice some peaks of whole numbers such as 1, 2, 3, 4 and 5. That makes sense because of the way people are going to rate movies and these movies probably only have few people watching them or just even one person watches them. Most movies distributed normally around 3-stars and 3.5 stars, and absolutely there is quite a number of outliers in 1-star movies. We observed in figure 3 that movies are more likely to be watched, those more ratings they got. That makes sense because good movies those are rated high, they have been watched by a lot of people.



**Figure 3.** relationship between the actual average rating and the number of ratings.

By applying item-based CF on the dataset, we built a model based on correlation between products themselves, starting with creating user-item matrix by using pivot table function. However, number of cells inside matrix is around 1,570,816, and level of sparsity is around 93.6%. Finally, to get a better prediction, we filtered out all the movies those have less than 100 reviews. Another method used to build recommender engine is Memory Based by computing Cosine Similarity model-based CF using SVD.

**7. Research finding and conclusion**

In this paper we have presented several techniques used to build recommender engines, such as content based and collaborative filtering technique. The major goal of recommender engines is to suggest new items to users like video, song, document, ....etc. We have reviewed many fields utilized recommender engines like financial services, Netflix and Amazon. We have shown content-based method that exploited user’s features or information about items to suggest new items, these features could be age, Sex, Job, nationality, genre of a movie. Also, we have covered collaborative filtering techniques based on rating history and tastes of similar users’ profile. We explored also some potential problems that appear in collaborative filtering like cold start, sparsity, scalability. Finally, we applied collaborative filtering techniques on the dataset that was imported from MovieLens website, these techniques are item-based CF by using correlation between items, Memory Based by computing Cosine Similarity, and Model Based by using SVD.

**Acknowledgments:** None.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- [1] M. Al-Emran, S. Zaza, and K. Shaalan, "Parsing modern standard Arabic using Treebank resources," in *2015 International Conference on Information and Communication Technology Research, ICTRC 2015*, 2015.
- [2] C. Mhamdi, M. Al-Emran, and S. A. Salloum, *Text mining and analytics: A case study from news channels posts on Facebook*, vol. 740. 2018.
- [3] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, 1997.
- [4] X. Amatriain and J. Basilico, "Recommender systems in industry: A netflix case study," in *Recommender systems handbook*, Springer, 2015, pp. 385–419.
- [5] M. Frey, *Netflix Recommends: Algorithms, Film Choice, and the History of Taste*. Univ of California Press, 2021.
- [6] M. D. Buhmann *et al.*, "Recommender Systems," in *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2011, pp. 829–838.
- [7] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3. Elsevier B.V., pp. 261–273, Nov-2015.
- [8] R. Burke, A. Felfernig, and M. H. Göker, "Recommender systems: An overview," *Ai Mag.*, vol. 32, no. 3, pp. 13–18, 2011.
- [9] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, 2002.
- [10] T. Tran and R. Cohen, "Hybrid recommender systems for electronic commerce," in *Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press*, 2000.
- [11] S. Zaza and M. Al-Emran, "Mining and exploration of credit cards data in UAE," in *Proceedings - 2015 5th International Conference on e-Learning, ECONF 2015*, 2015, pp. 275–279.
- [12] A. A. Saa, M. Al-Emran, and K. Shaalan, "Mining Student Information System Records to Predict Students' Academic Performance," in *International Conference on Advanced Machine Learning Technologies and Applications*, 2019, pp. 229–239.
- [13] S. A. Salloum, M. Al-Emran, S. Abdallah, and K. Shaalan, "Analyzing the Arab Gulf Newspapers Using Text Mining Techniques," in *International Conference on Advanced Intelligent Systems and Informatics*, 2017, pp. 396–405.
- [14] I. Arpacı *et al.*, "Analysis of Twitter Data Using Evolutionary Clustering during the COVID-19 Pandemic," *Comput. Mater. Contin.*, 2020.
- [15] M. Al-Emran, "Hierarchical Reinforcement Learning: A Survey," *Int. J. Comput. Digit. Syst.*, vol. 4, no. 2, pp. 137–143, 2015.
- [16] S. R. Naganna, B. H. Beyaztas, N. Bokde, and A. M. Armanuos, "ON THE EVALUATION OF THE GRADIENT TREE BOOSTING MODEL FOR GROUNDWATER LEVEL FORECASTING," *Knowledge-Based Eng. Sci.*, vol. 1, no. 1, pp. 48–57, 2020.
- [17] A. D. Mehr and O. Akdegirmen, "Estimation of Urban Imperviousness and its Impacts on Flashfloods in Gazipaşa, Turkey," *Knowledge-Based Eng. Sci.*, vol. 2, no. 1, pp. 9–17, 2021.
- [18] R. Tur and S. Yontem, "A Comparison of Soft Computing Methods for the Prediction of Wave Height Parameters," *Knowledge-Based Eng. Sci.*, vol. 2, no. 1, pp. 31–46, 2021.
- [19] A. Sharafati *et al.*, "The potential of novel data mining models for global solar radiation prediction," *Int. J. Environ. Sci. Technol.*, no. 0123456789, 2019.
- [20] Z. S. Khozani *et al.*, "Determination of compound channel apparent shear stress: application of novel data mining models," *J. Hydroinformatics*, 2019.
- [21] J. Sun and Y. Xie, "A web data mining framework for E-commerce recommender systems," in *Proceedings - 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009*, 2009.
- [22] A. Goel, P. Gupta, J. Sirois, D. Wang, A. Sharma, and S. Gurumurthy, "The who-to-follow system at Twitter: Strategy, algorithms, and revenue impact," *Interfaces (Providence)*, vol. 45, no. 1, pp. 98–107, Jan. 2015.
- [23] R. D. Lawrence, G. S. Almasi, V. Kotlyar, M. S. Viveros, and S. S. Duri, "Personalization of supermarket product recommendations," *Data Min. Knowl. Discov.*, vol. 5, no. 1–2, pp. 11–32, 2001.
- [24] P. C. Vaz, D. Martins De Matos, B. Martins, and P. Calado, "Improving a hybrid literary book recommendation system through author ranking," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, 2012.
- [25] "(PDF) A Hybrid Recommendation System."
- [26] M. J. Pazzani, "Framework for collaborative, content-based and demographic filtering," *Artif. Intell. Rev.*, vol. 13, no. 5, pp. 393–408, 1999.

- [27] “Content-Boosted Collaborative Filtering for Improved Recommendations.”
- [28] M. Balabanović and Y. Shoham, “Content-Based, Collaborative Recommendation,” *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.
- [29] K. Lang, “NewsWeeder: Learning to Filter Netnews,” in *Machine Learning Proceedings 1995*, Elsevier, 1995, pp. 331–339.
- [30] R. J. Mooney and L. Roy, “Content-based book recommending using learning for text categorization,” *Proc. ACM Int. Conf. Digit. Libr.*, pp. 195–204, 2000.
- [31] M. Pazzani and D. Billsus, “Learning and Revising User Profiles: The Identification of Interesting Web Sites,” *Mach. Learn.*, vol. 27, no. 3, pp. 313–331, 1997.
- [32] K. Al Fararni, F. Nafis, B. Aghoutane, A. Yahyaouy, J. Riffi, and A. Sabri, “Hybrid recommender system for tourism based on big data and AI: A conceptual framework,” *Big Data Min. Anal.*, vol. 4, no. 1, pp. 47–55, Mar. 2021.
- [33] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, “A literature review and classification of recommender systems research,” *Expert Syst. Appl.*, vol. 39, no. 11, pp. 10059–10072, Sep. 2012.
- [34] A. G. Prasad, M. K. Gourisaria, and L. K. Vashishtha, “Building hybrid recommendation system based on Hadoop framework,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 3493–3499.