Movie Recommendation System – Review and Comparision

Authors:

1. Trần Thị Hạnh:

Organization: National Economics University, Vietnam

Email: hanhtran122002@gmail.com

Mobile: 0973349510

2. Nguyễn Thị Hồng Anh:

Organization: National Economics University, Vietnam

Email: honganhng1711@gmail.com

Mobile: 0941529756

3. Phạm Văn Hồng:

Organization: National Economics University, Vietnam

Email: vanhong542002@gmail.com

Mobile: 0986824965

4. Mai Trà My:

Organization: National Economics University, Vietnam

Email: Mtmy330@gmail.com

Mobile: 0852871974

5. Lê Huy Phong

Organization: National Economics University, Vietnam

Email: phongleworktime@gmail.com

Mobile: 0982025911

Instructor: Nguyen Thi Quynh Giang – PhD

Organization: National Economics University, Vietnam

Email: giangntq@neu.edu.vn

Movie Recommendation System – Review and Comparision

Abstract

Movie recommendation systems play a pivotal impact in enhancing user experience by providing personalized content suggestions. This paper conducts a comprehensive review and comparison of two movie recommendation methods: content-based and collaborative filtering. The analysis uses data from the MovieLens dataset encompassing baseline model, followed by an exploration of content-based approaches utilizing movie attributes and lastly collaborative filtering including Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), and k-Nearest Neighbors (KNN) and their variants. The paper delves into the intricate details of each above method, examining their strengths and limitations. Furthermore, we suggest elements in a hybrid system, which is based on the outcomes of the best-performing models of content-based and collaborative approach to achieve a more robust and accurate recommendation system that need to be validated in real-world implementation.

Keywords: Movie recommendation systems, content-based, collaborative filtering, Singular Value Decomposition (SVD), k-Nearest Neighbors (KNN).

1. Introduction

Movies play a vital role in informing society about the various aspects prevailing in our world. They provide an escape from the real world, offer entertainment, and serve as a source of inspiration for many. Movies do not just entertain us, but they are also very effective in helping us organize our thoughts. Negative thoughts, anxiety and many other mental problems affect our health massively, without our knowledge. Instead of overthinking our thoughts, we should try letting things out by watching movies. For instance, Romantic and Comedy movies can help you face your real-life problems and might help you find a solution for it. According to a study by University of Maryland (2005), people who watched comedy movies found their blood to dilate by 22%. Besides, watching movies is also a way to bond with friends and family. They help us to connect better and make us laugh our hearts out. The joy of watching a good movie with family and friends is unparallel. There are different types of movies like some for entertainment, some for educational purposes, some are animated movies for children, and some are horror movies or action films. Movies can be easily differentiated through their genres like comedy, thriller, animation, action etc. Another way to distinguish among movies can be either by releasing year, language, director or by cast etc. Given the huge number of movies available all over the world, it is challenging for a user to find the appropriate movies suitable for his/her tastes. Different users like different movies or actors. It is important to find a method of filtering irrelevant movies and find a set of relevant movies. Movie Recommendation System helps us to search our preferred movies among all these different types of movies and hence reduce the trouble of spending a lot of time searching our favourable movies.

Recommendation system is a useful technology that can alleviate the problem of overload of information provided to users. It predicts the grade of items to be recommended to the user, creates a list of recommendation rankings for each user, and makes it possible to recommend items related to the user. These can be based on various criteria, including past rating, search history, demographic information, and other factors. Recommender systems are trained to understand the preferences, previous decisions, and characteristics of people and products using data gathered about their interactions. Many companies such as Netflix, YouTube, Amazon, employ recommendation systems to better serve their customers and raise their profits. But it is still a valuable study topic because finding what the users want from accessible resources is a difficult task, especially when our preferences change over time. Movie Recommendation System is an ML-based approach to filtering or predicting the users' movie preferences based on their past choices and behaviours. It's an advanced filtration mechanism that predicts the possible movie choices of the concerned user and their preferences towards a domain-specific movie. The Movie Recommendation System architecture is a complex process that utilizes various algorithms to suggest movies to users based on their preferences. The architecture involves collecting data on user behaviours and using that data to create a personalized list of suggestions.

According to M.ali Ghazanfar et al (2010) and AKTU et al (2020), there are some problems related to the recommendation system that reduce the effectiveness of these systems:

- *Cold-start problem:* is when a user registers for the first time, he/she has not watched any movie. Hence, the recommendation system does not have any user-item interactions based on which it can give result.
- *Data sparsity problem:* happens when the user has rated very few items based on which it is difficult for the recommendation system to give accurate results. In this problem, the results given are not very similar to the expected result. And data sparsity always leads to coverage problems.
- *Scalability:* the encoding goes linearly on items. The system works efficiently when the data set is of limited size. As the data set increases, it becomes difficult for the recommendation system to give accurate results based on varying genres of movies.

2. Literature review:

Over the past decade, a large number of recommendation systems for a variety of domains have been developed and are in use. According to Eyrun A. Eyjolfsdottir et al (2010), two main paradigms for the filtering are content-based approach and collaborative filtering approach.

2.1. Collaborative Filtering:

The idea of collaborative filtering was first introduced in 1991 by Goldberg et al. Since then, this method has grown in popularity and is frequently employed by e-commerce platforms such as Amazon, Drugstore, etc. Collaborative filtering (CF) is one of the most successful recommender techniques (Xiaoyuan Su et al, 2023). The basic idea of Collaborative Filtering (CF) algorithms is to make item recommendations or predictions based on the notions of other like-minded users. Such assessment is determined either explicitly or implicitly. In an explicit determination, users are asked to provide their ratings in a one to five scale, which are then utilized to measure the similarity. In an implicit determination, users' rating is established based on the browsing behaviours. However, if the item set is large and users rate a small fraction of these, it is often difficult to find similarities between users. This leads to low accuracy predictions or even to failure to make predictions. Besides, explaining ability is quite difficult when using collaborative filtering with latent-factor models or matrix factorization models. Y.B. Fernandez et al (2010) utilizes state of the art Matrix Factorization (MF) recommender systems: NMF, PNF, and proposes a new Explainable Matrix Factorization (EMF) technique that computes an accurate top N recommendation list of items that are explainable. The authors also introduce new explanation quality metrics called Mean Explainability Precision (MEP) and Mean Explainability Recall (MER). They got the average RMSE of EMF is 1.3411 by using cosine similarity.

2.2. Content-based Filtering:

Balabanovic et al (1997) had proposed a Content-based filtering recommendation system which can be applied in different domains such as books, movies, videos, or music. It uses different features such as author, genre, and most frequently used words. Content-based filtering is also known as cognitive filtering (H.Li et al, 2012). It recommends items based on the textual information of an item, under the assumption that users will like similar items to the ones they liked before. It provides recommendation to the user with unique taste. J. Son et al (2017) applied Content-based filtering for recommendation systems using multi-attribute networks in which a node of the network represents a movie, and a weighted edge represents the degree of relevance between two movies. Dice similarities are computed to determine the closeness between paired items.

2.3. State of the Art methods:

Various techniques for collaborative and content-based filtering are discussed in the literature survey. A comparison of some recent techniques used for recommendation is shown in the following table.

Table 2.3. Comparison of State-of-the-Art methods

Ref	**CF/CB**	**Dataset**	**Method**	**Similarity Measures**	**Performance**
J.Son et al, "Content-based filtering for recommendation systems using	CB-MN	Movie lens - 1M	CB-MN system, FW		Overspecialization and data sparsity is improved

multiattribute networks", 2017					
Y.B.Fernandez et al, "Providing Entertainment by Content-based Filtering and Semantic Reasoning in Intelligent Recommender Systems", 2008	CB	IMDB and BBC web server	SA		Overspecialization problem is solved and observed that when number of clusters increases accuracy decreases.
B.Abdollahi et al, "Explainable Matrix Factorization for Collaborative Filtering", 2016	CF	Movie lens	EMF, NMF, PMF	Cosine	Average RMSE of EMF is 1.3411
M.G.Vozalis et al, "Using SVD and demographic data for the enhancement of generalized Collaborative Filtering", 2007	CF	Movie lens	SVD, UBCF, IBCF	Demographic correlation	IBCF produces less error value than UBCF

H.Koohi et al, “User based Collaborative Filtering using fuzzy C-means”, 2016	CF	Movie lens with 6.3% rating available	K-means, SOM, Fuzzy clustering	Pearson Correlation	Fuzzy C-means and max average accuracy-80.44 and Fuzzy C-means and max Pearson accuracy-81.1

Through evaluating research on movie recommendation systems, our team found that Collaborative filtering approaches are commonly used in all systems. However, both Content-based filtering and Collaborative filtering approaches have certain limitations. With the Content-based filtering method, recommendations may be limited to items similar to those the user has already consumed. Hence user cannot discover new and diverse items is the biggest drawback. Because every year there will be many new movies released and this method only relies on movies that users have watched in the past to make suggestions. Moreover, if the user is a new user or has not seen any movies before, this method will not be able to make any recommendations, resulting in the user may miss out on the best movies. Regarding the Collaborative filtering method, the two biggest limitations of this method are the cold start problem, and that the accuracy of the suggestion can be affected by external factors such as: differences in preferences between users, number of users and number of movies in the database.

3. Dataset

Data overview

The dataset used in this project is a subset of the Full MovieLens Dataset which consists of movies released on or before July 2017. It has 2 main components: ratings and movies details, which are included in 5 files: metadata, links, keywords, credits, and rating.

Ratings have been obtained from the Official GroupLens website with a collection of 100,004 reviews by 671 users. The Movie Details, Credits and Keywords have been collected from the TMDB Open API. It includes information such as genres, cast, director, overview, keywords, production company.

Data preprocessing

Based on UserID, we divided the ratings dataset into test set and train set, using 75 percent of each user's reviews to train the models and 25 percent to validate the models. We accomplished the above utilizing the train test split method in scikit-learn library. Some cleaning processes included removing redundant, null rows and non-contributing features from the dataset have been done before merging metadata, keywords, and credits. We then extract and construct all necessary features such as movie description, weighted rating, cast, ... Lastly genres and other categorical features are label encoded for the purpose of representation.

Data analysis

The merged meta data about movie information includes 9082 movies with 16 features.

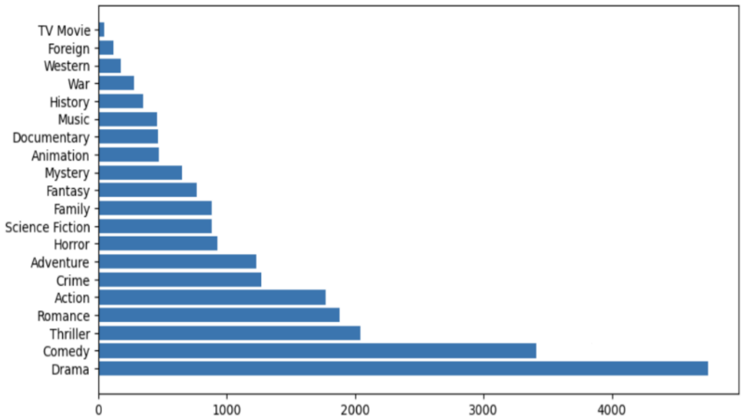


Figure 3.1: Genres Distribution

According to the graph above, there are 20 genres in total. Drama, comedy, thriller are the most popular genres in the dataset while TV movie, Foreign, Western being the least.

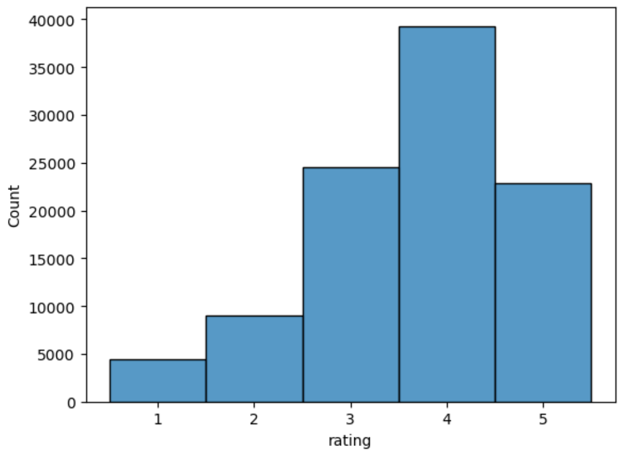


Figure 3.2: Ratings distrubution

Most movies are rated at 3.5 or higher, which implies an overall positive rating in Figure 3.2.

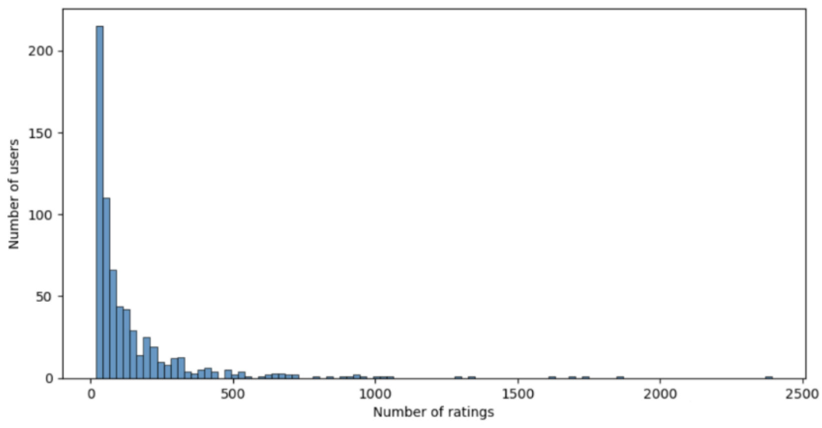


Figure 3.3: Number of ratings per user

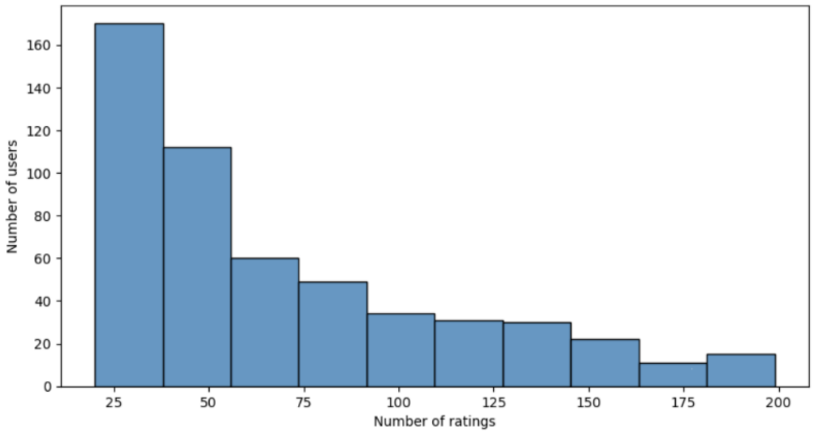


Figure 3.4: Number of ratings smaller than 200

From the two graphs above, we see that most users rate 25 - 50 movies and 97.62% of the users' rate less than 200 movies.

4. Methodology:

Baseline model

Baseline models offer generalized recommnendations to every user based on movie's popularity, movie release year, weighted rating (TMBD rating) and average rating of each movie in the rating file. The basic idea behind this recommender is that movies that are more popular, highly rated and released recently will have a higher probability of being liked by the average audience. This model does not give personalized recommendations based on the user. The implementation of this model is extremely trivial. All we have to do is sort our movies based on popularity, rating and release year and then display the top movies of our list. We try out between using weighted rating, which is already available in the metadata files and the average rating calculated in the rating file.

Content-based Filtering appoarch

```
smd[smd.title == 'The Dark Knight']
```

	id	movieId	title	description
6897	155	58559	The Dark Knight	Batman raises the stakes in his war on crime. ...

	id	movieId	title	description
132	414	153	Batman Forever	The Dark Knight of Gotham City confronts a das...
524	268	592	Batman	The Dark Knight of Gotham City begins his war ...
1113	364	1377	Batman Returns	Having defeated the Joker, Batman now faces th...
2578	14919	3213	Batman: Mask of the Phantasm	An old flame of Bruce Wayne's strolls into tow...
2695	820	3386	JFK	New Orleans District Attorney Jim Garrison dis...
7561	40662	79274	Batman: Under the Red Hood	Batman faces his ultimate challenge as the mys...
7896	69735	90603	Batman: Year One	Two men come to Gotham City: Bruce Wayne after...
7926	49026	91529	The Dark Knight Rises	Following the death of District Attorney Harve...
8160	123025	98124	Batman: The Dark Knight Returns, Part 1	Batman has not been seen for ten years. A new ...

Figure 4.1: Similar movies for 'The Dark Knight' by 'description'

The model works on the idea of finding similar items based on the content of items. In the case of movies, we tried to include the main features of movies like genres, description, and some other features such as production company, spoken languages. With these features, predictions are made for a particular user based on movies they have watched. For example, we recommend movies similar to 'The Dark Night' using movie description in Figure 4.1. The result of top 10 similar movies is quite reliable. Cosine-similarity is the similarity measurement between movies, below is the formular we use.

$$\cos(\theta) = \frac{A\ .\ B}{||A||\ ||B||} = \frac{a}{\sqrt{\sum_i A_i^2}\ \sqrt{\sum_i B_i^2}}$$

Collaborative Filtering appoarch

The underlying idea of collaborative filtering is that if a user A has similar preferences to a user B on certain items, then A is likely to have similar preferences to B on other items as well. This approach assumes that users who agreed in the past tend to agree again in the future. Collaborative filtering does not rely on explicit knowledge about items or users. Instead, it focuses on finding patterns and similarities based on user behaviours or preferences. In this project we consider some models in Table 2.3. The Surprise library is used, and model selection is based on the RMSE on the validation set (size = 20%). After choosing the best 4 models, we combine them linearly to create a new model with better performance. We then predict the ratings for anti-train set, which consists of the movies that users have not rated and evaluate these 5 models in the test set by the self-built metric.

Table 4.1. Collaborative filtering algorithms

Algorithm	Description
KNNBasic	A basic collaborative filtering algorithm which takes the maximum number of neighbours k to consider and the similarity metric as parameters
KNNwithMeans	A basic collaborative filtering algorithm similar to KNNBasic which considers the mean ratings of each user
KNNWithZScore	A basic collaborative filtering algorithm similar to KNNBasic which considers the z-score normalization of each user
KNNBaseline	A collaborative filtering algorithm which considers a baseline rating
SVD	A method of decomposing a matrix, usually called A, into three smaller matrices: U, S, and V

SVDpp	An extension of SVD that considers implicit ratings
NMF	A matrix factorization method that factorizes a matrix A n into two matrices W and H and these two matrices only contain non-negative elements
Baseline	Algorithm predicting the baseline estimate for given user and item. $\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i$
Slope One	An algorithm to create a linear relation between items preferences to handle data sparsity

Evaluation

Based on characteristics of movie recommendation system we want to maximize the number of recommendations (movie predictions) that user will later watch. Hence, metric that is similar to precision will be most suitable for model evaluation. For collaborative filtering, we predict rating of *movieId* in the test set and then use RMSE, MAE to evaluate the performance.

On the other hand, we need one general metric for all three approaches including baseline and content-based model. Metric is measured by the average percentage of movie recommendations that are actually watched by users in the test set. For collaborative, we will predict rating of all the movies that users have not watched before, sort descending and recommend the top N movies. Below is the formular of the customized metric:

$$\frac{1}{Number\ of\ users}\sum_{i}\frac{Count(Movies_{recommend_i} == Movies_{test_i})}{Count(Movies_{recommend_i})}$$

5. Result

Baseline Model

By experimenting with different features, we have found using popularity and average rating (in the rating file) witnessing a higher result 2.3% (model 2, 3) compared to 0.179% of movies recommended watched by user using weighted rating (model 1) Table 5.1. The results also point out that including the release year (Baseline Model 3) does not help in increasing the performance. Since Baseline 2 (the best result) using "Popularity" & "Average Rating", we will consider trying to make use of this to enhance the performance of the next models.

Table 5.1: Baseline model results

Model	Result
Baseline 1	0.00179
Baseline 2	0.023
Baseline 3	0.023

Content-based Filtering

Content-based filtering involves two steps. First, we assume that among all the string type attributes related to movies, users are only interested in some of that. Therefore, feature selection needs to be conducted. By using each feature at a time, we found that production company and keywords bring extraordinary results based on the customized metric. Next step, those features will then be concatenated with embedded movie description. Even though we found movie features with higher impact in the previous step, it is still necessary to experiment with different combinations of movie features and movies description.

Table 5.2: Combination of Movie Feature Results

	TF-IDF	**CountVectorizer**
description + genres	0.041	0.036
keywords + director + genres + cast	0.035	0.052
description + production_companies + keywords	0.076	0.066
description + all text features	0.091	0.089

Performance showing in Table 5.2 suggest that the model performs better when using all text features including 'title', 'description', 'genres', 'cast', 'production_companies', 'keywords', 'director', 'spoken_languages', 'production_countries' and is recommended based on TF-IDF. It was noted that using only 'description' and 'genres' yields poor results. However, when replacing 'genres' with 'production_companies' and 'keywords,' there is a significant improvement in the results. One of the possible reasons for this is that generic features like "genres" will make it more difficult for the model to distinguish accurately than the descriptive feature "keywords" when in case the model must distinguish between films of the same genre but the nature of the content of those films is different. Therefore, it can be said that one of the two features, 'production_companies' or "keywords," is quite important when using the Content-based method.

Collaborative Filtering

We experimented with SVD, NMF, KNN, and their variations. Table 5.3 shows results of best performance algorithm in terms of RMSE and MAE: SVD, SVDpp, KNN Baseline and Baseline. Particularly, KNN Baseline has a slightly higher RMSE compared to the other three algorithms (about 0.01 higher).

Table 5.3: Collaborative Filtering Best Results

Algorithm	RMSE	MAE
SVD	0.900	0.696
SVD pp	0.894	0.688
KNN Baseline	0.906	0.696
Baseline Only	0.896	0.695
Combined	0.883	0.681

In the pursuit of optimization, a linear combination approach was employed by integrating four prominent algorithms - SVD, KNN Baseline, SVDpp, and Baseline. By assigning specific weights to each algorithm, we experimented with different weight combinations and found the result of assigning SVD, KNN Baseline, SVDpp, and Baseline with a weight of 0.25, 0.25, 0.3, and 0.2, has the best outcome. The analysis reveals a decrease in both RMSE and MAE when compared to the outcomes of each individual algorithm Table 5.3. This underscores the intricate nature of algorithmic integration and the need for a nuanced approach in enhancing recommendation systems.

Evaluation on the test set using the customized metric is conducted by re-training the model that performed well on the validation sets. Result of this approach yields relatively low results, as shown in Table 5.4

Table 5.4: Result of collaborative algorithms on test set

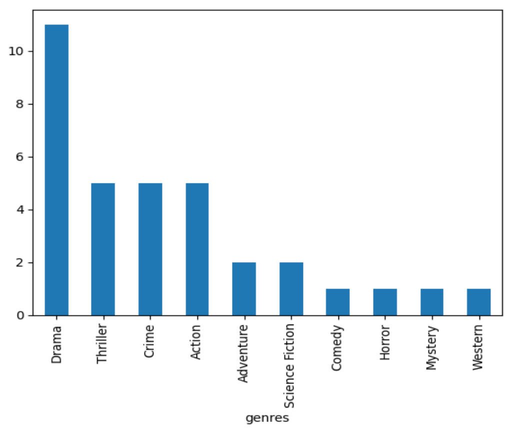
Algorithm	Result
SVD	0.0634
SVDpp	0.0684
Baseline only	0.0819
KNN Baseline	0.00134
Combined	0.0651

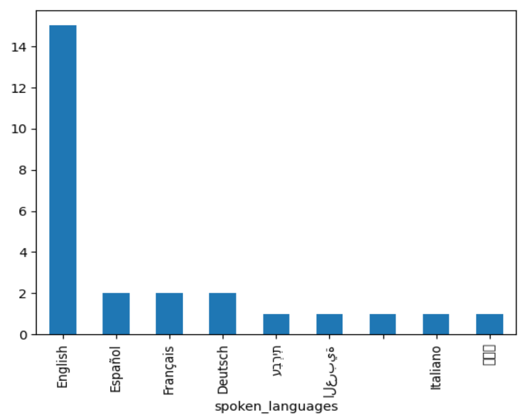
Comparison example of two approaches

We take user with *id 583* as an example since this user has relatively a small number of movies. He typically watched drama and types of thriller/action/crime movies. Most of the movies are in English, some in French/Espanol/Deutsch. He may prefer Warner Bros and Syncopy over other production companies.

	title	description
266	Pulp Fiction	A burger-loving hit man, his philosophical par...
284	The Shawshank Redemption	Framed in the 1940s for the double murder of h...
972	Raiders of the Lost Ark	When Dr. Indiana Jones – the tweed-suited prof...
986	GoodFellas	The true story of Henry Hill, a half-Irish, ha...
1369	Good Will Hunting	Will Hunting has a genius-level IQ but chooses...
1860	American History X	Derek Vineyard is paroled after serving 3 year...
2304	American Beauty	Lester Burnham, a depressed suburban father in...
2390	Fight Club	A ticking-time-bomb insomniac and a slippery s...
6819	Superbad	High school best buddies are facing separation...
6905	I Am Legend	Robert Neville is a scientist who was unable t...
6981	The Dark Knight	Batman raises the stakes in his war on crime. ...
7648	Inception	Cobb, a skilled thief who commits corporate es...
8031	The Dark Knight Rises	Following the death of District Attorney Harve...
8310	Django Unchained	With the help of a German bounty hunter, a fre...

Figure 5.1: Movies UserId 583 watched





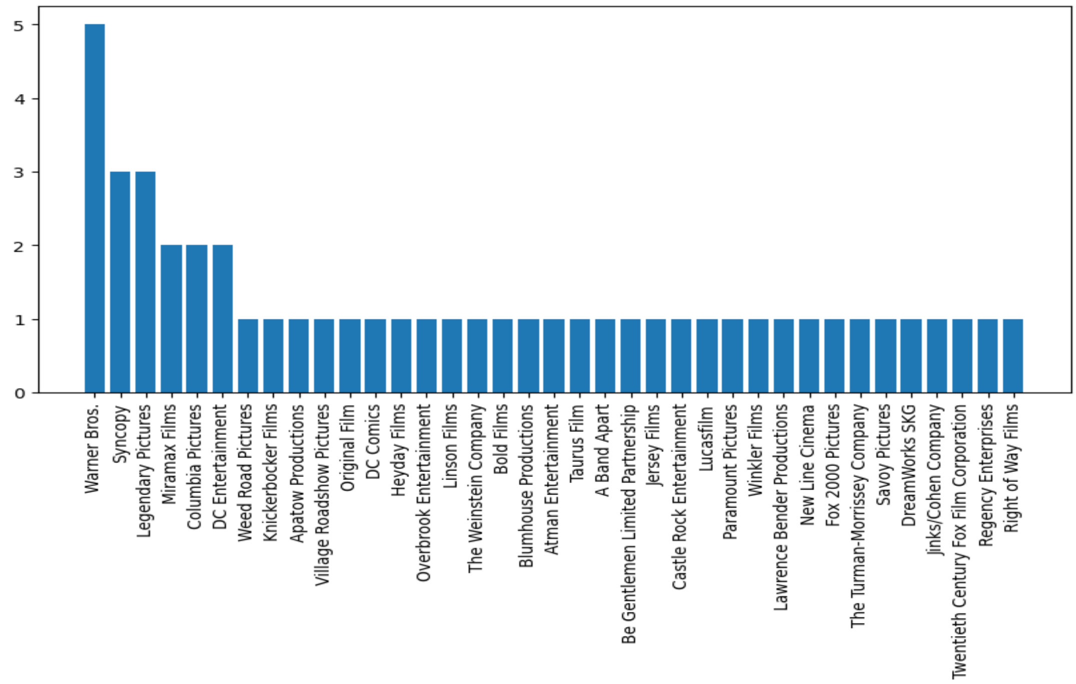


Figure 5.2: Some movie attributes distribution

Suggested films based on Content-Based:

	title	description
132	Batman Forever	The Dark Knight of Gotham City confronts a das...
1062	Indiana Jones and the Last Crusade	When Dr. Henry Jones Sr. suddenly goes missing...
1134	Batman Returns	Having defeated the Joker, Batman now faces th...
1692	Indiana Jones and the Temple of Doom	After arriving in India, Indiana Jones is aske...
2599	Batman: Mask of the Phantasm	An old flame of Bruce Wayne's strolls into tow...
6218	Batman Begins	Driven by tragedy, billionaire Bruce Wayne ded...
7024	Indiana Jones and the Kingdom of the Crystal S...	Set during the Cold War, the Soviets – led by ...
7659	Batman: Under the Red Hood	Batman faces his ultimate challenge as the mys...
8265	Batman: The Dark Knight Returns, Part 1	Batman has not been seen for ten years. A new ...
8334	Batman: The Dark Knight Returns, Part 2	Batman has stopped the reign of terror that Th...

Figure 5.3: Movies recommended for user 583 – Content-based

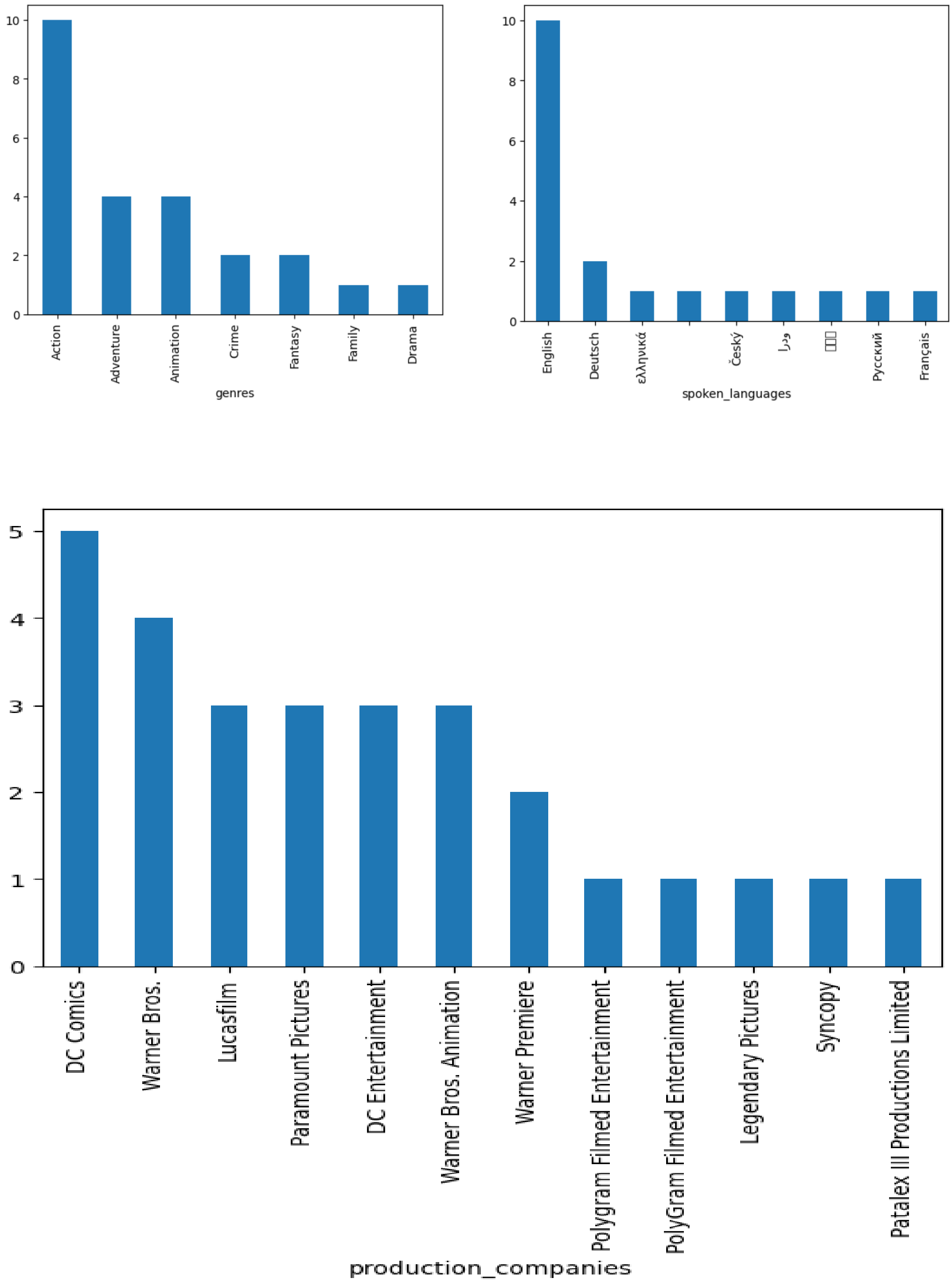
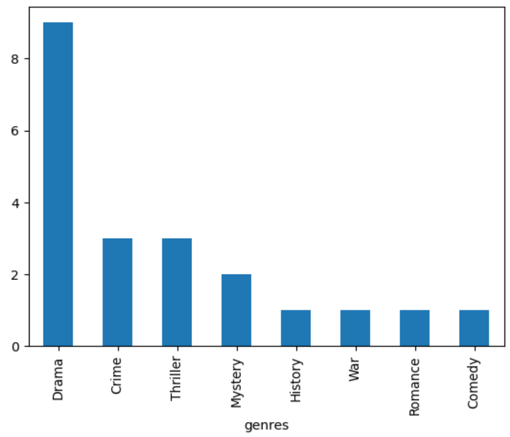


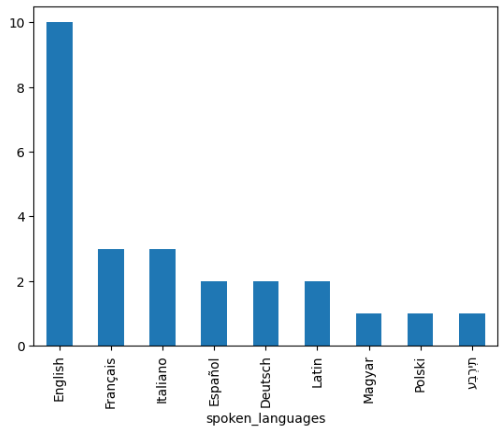
Figure 5.4: Movie attributes distribution – Content-based

Suggested films based on Collaborative filtering (Baseline Only):

	title	description
48	The Usual Suspects	Held in an L.A. interrogation room, Verbal Kin...
472	Schindler's List	The true story of how businessman Oskar Schind...
699	The Godfather	Spanning the years 1945 to 1955, a chronicle o...
736	Rear Window	Professional photographer L.B. "Jeff" Jeffries...
740	North by Northwest	Advertising man Roger Thornhill is mistaken fo...
744	Casablanca	In Casablanca, Morocco in December 1941, a cyn...
757	All About Eve	From the moment she glimpses her idol at the s...
968	One Flew Over the Cuckoo's Nest	While serving time for insanity at a state men...
994	The Godfather: Part II	In the continuing saga of the Corleone crime f...
2780	Modern Times	The Tramp struggles to live in modern industri...

Figure 5.5: Movies recommended by Collaborative filtering





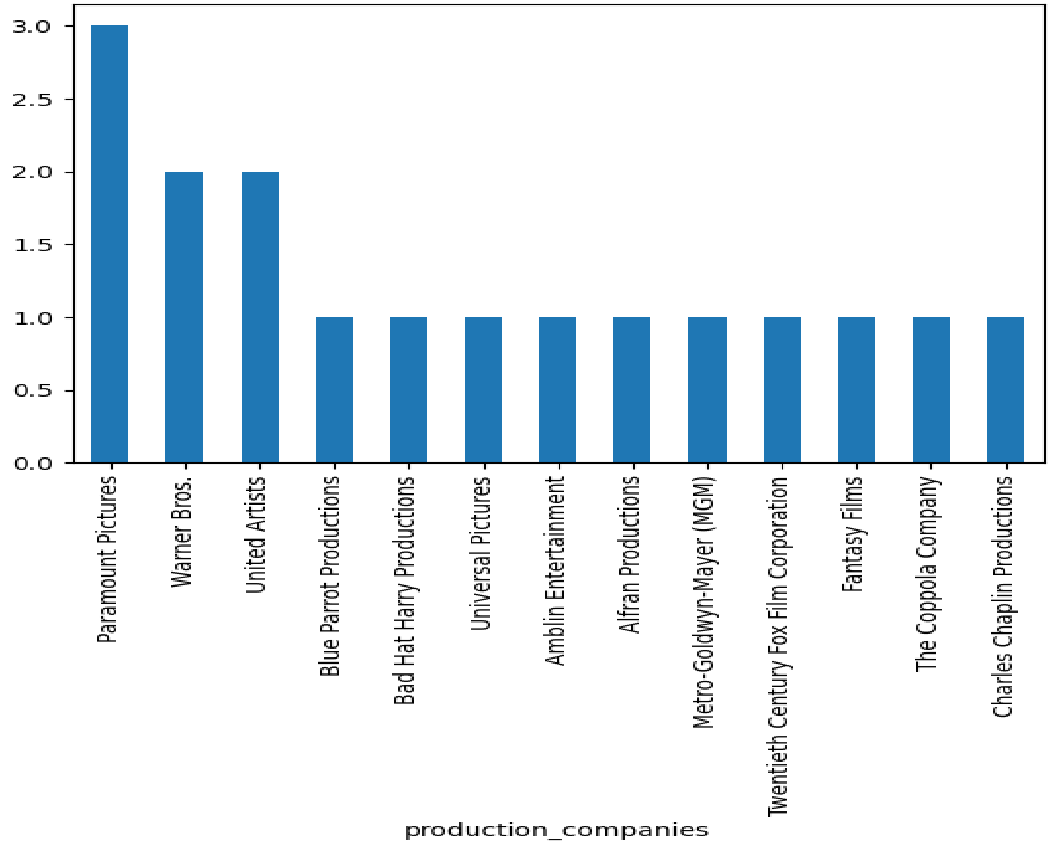


Figure 5.6: Movie attributes distribution – Collaborative filtering

Based on the movie attributes such as 'genres', 'spoken_languages', 'production_companies', Collaborative Filtering model is producing recommendations quite relative to user's preferences. However, noticed that the watching history of the user contains 2 seasons of the same series 'The Dark Knight', which means he is especially interested in this series, Collaborative Filtering cannot capture that while Content-Based approach recommends 7 out of 10 movies related to this series. From this example, we can see that Content-Based approach suggests significantly relevant movies, but it may limit user to discover other potential movies and other genres as well. On the other hand, Collaborative-Filtering maintains to offer a variety of kinds of movies that are quite relevant to user's preferences.

6. Conclusion:

Content-Based model and Baseline model are decent models for generating recommendations and understanding the notion of similarity. The investigation has highlighted that certain movie features significantly influence the performance of the recommendation system, with a notable impact from all text features. However, they lack diversity when recommending different types of movies for a given user. By the example above, 7 out of 10 movies of the same series are recommended, which means the slots for other relevant movies will be limited. Besides, Collaborative filtering with Baseline Only or KNN/SVD algorithms can capture the user related features well. Although Content-Based recommends movies that are closer in content, Collaborative Filtering can help users discover an overall more diversely similar movies that they might like in the future.

7. Future of Work:

Moving forward, it is recommended to conduct practical implementations, specifically deploying A/B tests, to precisely evaluate the standardized approach. A/B testing will provide a robust framework to assess the actual effectiveness of different methodologies in real-world scenarios, thereby offering valuable insights into the most reliable approach. Additionally, future work should consider the development of a hybrid system that integrates both content-based and collaborative filtering methods. Leveraging the strengths of various features and algorithms, as identified in this study, can enhance the overall recommendation system, catering to diverse user preferences and ensuring a more personalized and effective movie recommendation experience. This hybrid approach represents a promising avenue for future research and application in the field of movie recommendation systems.

References:

University of Maryland Medical Center, “University Of Maryland School Shows Laughter Helps Blood Vessels Fucntion Better”, ScienceDaily, March 19, 2005, Available: https://www.sciencedaily.com/releases/2005/03/050309111444.htm

M. ali Ghazanfar and A. Prugel-Bennett, “An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering,” *Lecture Notes in Engineering and Computer Science*, vol. 2180, Mar. 2010.

Inderprastha Engineering College, AKTU et al., “Movie Recommendation System using Cosine Similarity and KNN,” IJEAT, vol. 9, no. 5, pp. 556–559, Jun. 2020, doi: 10.35940/ijeat. E9666.069520.

D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992, doi: 10.1145/138859.138867.

“A Survey of Collaborative Filtering Techniques.” Accessed: Dec. 10, 2023. [Online]. Available: https://www.hindawi.com/journals/aai/2009/421425/

M. Balabanović and Y. Shoham, “Fab: content-based, collaborative recommendation,” *Commun. ACM*, vol. 40, no. 3, pp. 66–72, Mar. 1997, doi:10.1145/245108.245124.

H. Li, F. Cai, and Z. Liao, “Content-Based Filtering Recommendation Algorithm Using HMM,” in *2012 Fourth International Conference on Computational and Information Sciences*, Aug. 2012, pp. 275–277. doi: 10.1109/ICCIS.2012.112.

J. Son and S. B. Kim, “Content-based filtering for recommendation systems using multiattribute networks,” *Expert Systems with Applications*, vol. 89, pp. 404–412, Dec. 2017, doi: 10.1016/j.eswa.2017.08.008.

Y. B. Fernandez, J. J. P. Aris, A. G. Dolla, M. R. Cabrer, and M. L. Nores,” Providing Entertainment by Content-based Filtering and Semantic Reasoning in Intelligent Recommender Systems”, IEEE Transactions on Consumer Electronics 54(2):727-735, 2008.

B. Abdollahi and O. Nasraoui, “Explainable Matrix Factorization for Collaborative Filtering,” Apr. 2016. doi: 10.1145/2872518.2889405.

M. G. Vozalis and K. G. Margaritis,” Using SVD and demographic data for the enhancement of generalized Collaborative Filtering”, Information Sciences 177(15):3017-3037, 2007.

H. Koohi and K. Kiani,” User based Collaborative Filtering using fuzzy C-means”, Measurement: Journal of the International Measurement Confederation 91:134-139, 2016.

H. Ko, S. Lee, Y. Park, and A. Choi, “A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields,” *Electronics*, vol. 11, p. 141, Jan. 2022, doi: 10.3390/electronics11010141.

Eyrun A. Eyjolfsdottir, Gaurangi Tilak, Nan Li, “MovieGEN: A Movie Recommendation System”, UC Santa Barbara: Technical Report, 2010.