International Journal of Applied Research

ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2018; 4(11): 306-315
www.allresearchjournal.com
Received: 12-09-2018
Accepted: 18-10-2018

Vivek Krishna
CSE Dept, Accurate Institute of Management & Technology, Greater Noida, Uttar Pradesh, India

Manish Singh
CSE Dept, Accurate Institute of Management & Technology, Greater Noida, Uttar Pradesh, India

Correspondence
Vivek Krishna
CSE Dept, Accurate Institute of Management & Technology, Greater Noida, Uttar Pradesh, India

Personalized movie recommendation system: Tailoring cinematic suggestions

Vivek Krishna and Manish Singh

DOI: https://doi.org/10.22271/allresearch.2018.v4.i11d.11462

Abstract

Recommender schemes are individual of the most profitable and off-course movements of machine education electronics in trade. This is an information draining approach namely used to foretell the inclination of that stoner. The most common fields place the recommender system is used are books, revelation, documents, music, videos, pictures, etc. In this the paper we've projected a film advice system that is to say established a unified filtering approach that form use of the facts given by druggie's studies them, and further advises the pictures that are best adapted to the stoner at another time.

The urged picture list is sifted according to the environments are likely to these pictures by old druggies and uses colourful engine-knowledge algorithms for this purpose. It also helps druggies to find the pictures of their choice established the videotape happening of different druggies in a direct and effective method outside destroying main show up useless scanning. The bestowed recommender scheme creates approvals using colourful types of information and dossier about druggies from the show dataset. The exact can again browse the pieces of advice without difficulty and find a videotape of their choice.

Recommender System is a fashion that's used to approve a part or product to a decent established the muted silver in color's preference'. unified winnowing is an approach that's widely used in recommender orders. Item- article-based unified penetrating is a unified filtering recommender scheme fashion place the mannerly got the approval established the correspondence among the part environments. Then, we present an approach place we calculate the likeness with the particulars established the sort of details. Any item concedes possibility concern further than individual kidney or order. Based on details' inclination to a particular kidney or order we intend a new part- article- based correspondence rhythmical.

Keywords: Recommendation systems, collaborative filtering (Item-based and Genre-based), Machine learning, content-based filtering user-based recommendations

Introduction

A approval structure is a type of news dribbling a system that attempts to foretell the priorities of a mannerly, and form hints established these preferences. There are a wide variety of operations for recommendation systems. These have come decreasingly popular over the last many times and are now employed in utmost online platforms that we use. The content of similar platforms varies from pictures, music, books, and videos, to musketeers and stories on social media platforms, products one-commerce websites, to people on professional and dating websites, to search results returned on Google. frequently, these systems are suitable to collect information about a stoner's choices and can use this information to ameliorate their suggestions in the future.

For illustration, Facebook can cover your commerce with colour stories on your feed in order to learn what types of stories appeal to you. occasionally, the recommender systems can make advancements based on the conditioning of a large number of people. For illustration, if Amazon observes that a large number of guests who buy the rearmost Apple MacBook also buy a USB- C- to USB appendage, they can recommend the appendage to a new stoner who has just added a MacBook to his wain. Two main approaches are extensively used for recommender systems. One is happy- based filtering, where we try to profile the stoner's interests using information collected, and recommend particulars based on that profile. The other is cooperative filtering, where we try to group analogous druggies together and use information about the group to make recommendations to the stoner. Movie recommendation systems have come an essential part of online movie streaming platforms. These systems help druggies in discovering pictures that match their preferences and interests. Deep literacy algorithms have shown pledge in the field of recommendation systems. In this paper, we explore the use of deep literacy in creating a movie recommendation system. Specifically, we use a type of deep literacy algorithm called cooperative filtering.

Challenges faced

The Recommendation System has a number of obstacles. These difficulties include the Cold Start issue, data sparsity, and scalability.

Cold Start Problem: It demands a enough number of consumers in the system to establish a counterpart. For example, if we wish to find a corresponding individual or object, we competition them accompanying the free consumers or amount. At the beginning, a new consumer's profile is empty cause he has not ranked some part and the system does misunderstand about welcome inclinations, making it intolerable for bureaucracy to present him with pieces of advice on some article. The unchanging may be pronounced about a new part, that has not happened inspected by any consumer because it is new to the consumer. Both of these issues concede possibility be handled by engaging composite approaches. Data Sparsity: The consumer or grade cast contains rather little dossier. It is troublesome to find people the one has ranked the unchanging belongings cause the majority of consumers do not review the parts. As a result, verdict a group of family who score current fashion enhances troublesome. When skilled is little news on a user, making approvals enhances intensely troublesome. Scalability: Collaborative filtering uses big amounts of dossier to correct dependability, which makes necessary a most of possessions. This Big dossier question causes processing to enhance harmful and vague as facts expands rapidly.

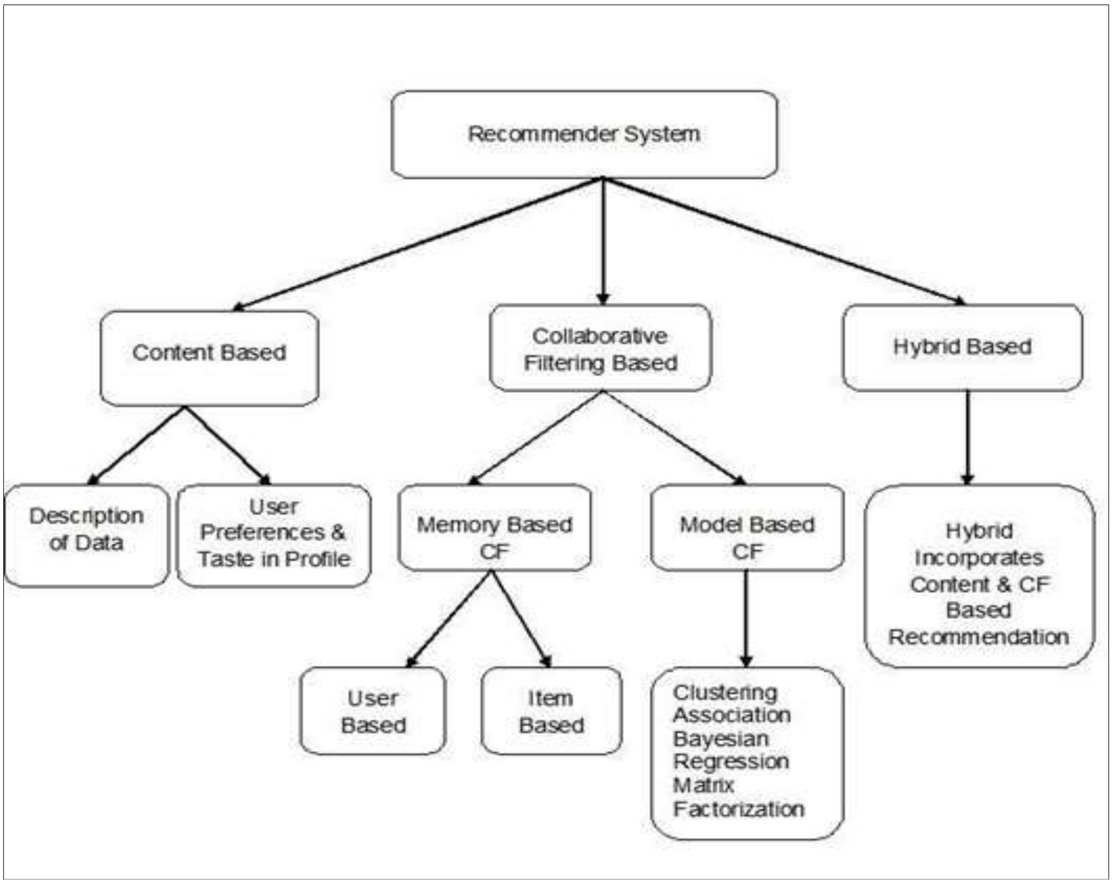


Fig 1: Classification of Recommendation Systems

Background

Content-located penetrating, collaborative refining, and mixture filtering are three types of approval methods. Content-located filtering includes resolving the lineaments of movies and making approvals established the similarity 'tween the facial characteristics of the cinema and the user's priorities. Collaborative percolating includes analyzing the act of consumers and making recommendations established the correspondence betwixt the behavior of the consumers and the consumer's predilections. Hybrid filtering connects two together content-based and cooperative permeating to create recommendations. Deep knowledge models have proved great promise entirely three types of cleaning.

Collaborative filtering (Item-based and Genre-based)

Collaborative filtering is a type of recommendation algorithm that relies on the conditions and preferences of other druggies to make recommendations. The beginning idea is that druggies with analogous tastes will like analogous pictures. There are two ways to enforce collaborative filtering: item-based and stoner-based. A collaborative filtering system that is centered around stoners suggests images to stoners based on the conditions of stoners who are similar to them

Collaborative filtering based on items suggests images that are similar to images that the stoner has previously rated highly.

Item-located and type-based cooperative winnowing are two popular approaches second-hand in film advice systems established machine intelligence. Item-based cooperative percolating advises movies to a consumer established the ratings of other consumers the one have ranked the same features. The fundamental idea is that if two consumers have ranked the unchanging set of movies likewise, before they are likely to have comparable weaknesses in flicks. The system first computes the similarity 'tween all pairs of flicks based on the ratings likely apiece consumers. Then, it recommends films that are related to the ones the consumer has before ranked highly.

On the other hand, genre-based collaborative filtering recommends movies to a user based on their preference for a specific genre of movies. In this approach, the system identifies the genres that the user has previously enjoyed and recommends movies from those genres. For example, if a user has watched and enjoyed several romantic comedies, the system will recommend other romantic comedies.

Both item-based and genre-based collaborative filtering have their strengths and weaknesses. Item-based filtering tends to perform well for users who have rated many movies, while genre-based filtering works well for users who have a strong preference for specific movie genres. Therefore, a combination of both approaches can be used to create a more effective recommendation system.

Machine learning algorithms, such as neural networks and matrix factorization, can be used to implement these approaches in a movie recommendation system. By analysing a user's movie viewing history and preferences, the system can make accurate predictions about the movies the user is likely to enjoy, thereby improving the user's overall movie-watching experience.

Machine Learning

Machine learning plays a crucial role in the development and improvement of movie recommendation systems. The goal of these systems is to personalize the movie-watching experience for each user by recommending movies that they are likely to enjoy. Machine learning algorithms are used to analyse the vast amounts of data that are available, including movie metadata, user ratings, and viewing histories, in order to make accurate predictions about which movies the user will like.

One common approach second hand in movie advice schemes is cooperative filtering. Collaborative leaking depends the preferences of added consumers the one have similar tastes to create approvals. Machine learning algorithms maybe used to analyse this data and label patterns in consumer behavior, such as correspondences in picture inclinations, which can before be used to form embodied recommendations. Another approach is content-located percolating, which uses the traits of the pictures, in the way that genre, cast, and plot, to create approvals. Machine learning algorithms maybe used to analyse this data and label patterns in flick characteristics, that can before be used to approve movies to consumers accompanying comparable tastes.

Deep learning algorithms, such as neural networks and matrix factorization, have also shown promise in movie recommendation systems. These algorithms can analyse the data in more depth, making use of complex patterns and relationships between movies and users to make even more accurate recommendations.

In addition to these approaches, machine learning algorithms can also be used to optimize the performance of recommendation systems over time. For example, by listening consumer interactions accompanying bureaucracy, machine intelligence algorithms can learn to develop the pertinence of recommendations and fit to changes in consumer behaviour.

Overall, machine learning plays a critical role in movie recommendation systems, enabling these systems to make personalized and accurate recommendations to users, improving their movie-watching experience, and increasing user engagement with the platform.

Deep Learning

Deep learning plays a significant role in improving the performance of movie recommendation systems. Traditional recommendation systems use machine learning algorithms such as collaborative filtering to predict user preferences. However, these methods have limitations in terms of scalability and accuracy.

Deep learning models can overcome these limitations by learning complex patterns and relationships between users and movies. These models can analyze large amounts of data and make accurate predictions, even for users with sparse data.

The following are some of the ways in which deep learning can be used in movie recommendation systems:

Feature extraction: Deep learning models can be used to extract features from movies and users. For example, a convolutional neural network (CNN) can analyze movie posters and extract visual features such as color, texture, and shape. Similarly, a recurrent neural network (RNN) can analyze user behavior and extract temporal features such as the sequence of movies watched and the time of day.

Collaborative filtering: Deep learning models can be used to improve collaborative filtering by learning user-item interactions. For example, a matrix factorization model can learn the latent factors that describe the preferences of users and the attributes of movies. These factors can then be used to make personalized recommendations.

Hybrid filtering: Deep knowledge models can be used to integrate content-located and collaborative winnowing to create hybrid approvals. For example, a hybrid model can use a CNN to extract visual features from movie posters and a matrix factorization model to learn user-item interactions.

Sequential modeling: Deep learning models can be used to model sequential data, such as the sequence of movies watched by a user. For example, an RNN can be used to predict the next movie that a user is likely to watch based on their viewing history.

In summary, deep knowledge plays a important role in reconstructing the veracity and scalability of movie approval plans. By using deep knowledge models to extract looks, learn consumer-part interactions, and model subsequent dossier, recommendation wholes can support personalized and appropriate pieces of advice to users, superior to revised user date and delight.

Content-Based Filtering

Another coarse approach when plotting recommender systems are satisfied- located cleaning. Filtering methods based on content are established a writing of the article and a description of the stoner's advantages. These styles are best adapted to positions where skillet's famous dossier on an article (name, position, description.), but gone the accurate. Content- located recommenders treat pieces of advice as a stoner-distinguishing classification question and learn a classifier for the manner less enjoys and dislike established a part's features. Throughout, keywords are utilized to highlight the specifics, and a precise profile is created to represent the kind of article this stoner selects. Put differently, these algorithms attempt to validate details that bear resemblance to those that a stoner finds appealing in their experiences or is currently examining. A polite sign-in medium is not necessary to support this constantly impromptu sketch. Specifically, outstanding applicant details are compared, and the corresponding details are suggested based on a preliminary ranking of the accurate and best choice. This denotes an information-piercing survey and the restoration of established facts.

One common method in movie recommendation systems is content-based filtering, which bases recommendations on the qualities of the images. The beginning idea is that if a stoner has enjoyed certain pictures in the history, they're likely to enjoy pictures with analogous characteristics.

The part of content- based filtering in movie recommendation systems is to identify the features or characteristics of the pictures that are important to druggies and to use this information to make individualized recommendations. These features can include movie stripes, actors, directors, plot summaries, or keywords. The system analyzes the characteristics of the pictures that the stoner has watched and liked in the history and also recommends pictures with analogous features.

User-Based Recommendations

User-based recommendations is an approach used in movie recommendation systems that relies on the preferences of other users who have similar tastes to make recommendations. The underlying idea is that if two users have similar tastes in movies, then they are likely to enjoy the same movies.

The role of user-based recommendations in movie recommendation systems is to identify users with similar tastes and to use this information to make personalized recommendations. The system analyzes the ratings and viewing histories of users and identifies other users with similar preferences. Next, it suggests movies that these comparable users have given high ratings to but which the user in question hasn't seen or rated yet.

User-based recommendations have several advantages over other approaches such as content-based filtering. Firstly, it can capture the nuances of user preferences and recommend movies that are outside the user's usual preferences but still similar to movies they have enjoyed in the past. Secondly, it can help users discover new movies and genres that they may not have heard of before but are popular among users with similar tastes.

However, user-based recommendations also have their limitations. For example, they require a significant history of ratings and interactions with the system to make recommendations accurately. Additionally, it concede possibility struggle to advise movies that are outside the consumer's typical predilections or that have not happened well ranked by other identical consumers.

Machine learning algorithms, such as collaborative filtering, can be used to implement user-based recommendations in movie recommendation systems. These algorithms can analyze the data on user ratings and viewing histories to identify patterns and make more accurate recommendations over time. The system can also be optimized over time, using machine learning algorithms to identify users with similar tastes and to make more accurate recommendations.

Overall, consumer-based pieces of advice are an main approach in movie approval systems, providing consumers with embodied approvals based on the inclinations of other consumers accompanying similar tastes.

Working/Methodology

Data-Set Description

We have taken two data-sets from two different sources:

1. IMDB 5000 Movie Dataset: It contains a total of 4806 movies and a total of 23 columns about its description and it has two files' movies and credits. And a total of 35different languages. We used this dataset for a content-based recommender system.
2. Dataset Movie lens (ml-most recent tiny) 600 things used 100,000 ratings and 3,600 tag applications to 9,000 films. We second hand this dataset for genre-located part-item collaborative cleaning, and for article based cooperative filtering.

Solution Approach

We've used a content- located approval method i.e. Using an IMDb dataset of 5000 pictures. We're utilizing the Count Vectorizer, and calculating the fleck Product will straightforwardly present us the Cosine Similarity Score First we've permeated and fashioned the dossier convenient for point beginning because we can work on it. We immediately have a pairwise cosine likeness mould for all the pictures in our dataset. again, we find ultimate agreeing pictures established the cosine likeness score We take all features identical as cast, company, manager, and kind, that decide the content of a flick in a text line. One element that we notice about our recommendation whole is that it approves pictures anyway of environments and celebrity. We've used an article- located cooperative advice structure utilizing the ml- last limited dataset and requesting equating we find the correspondence between details and approve the pictures whose equivalence worth is more.

Language and Libraries Use

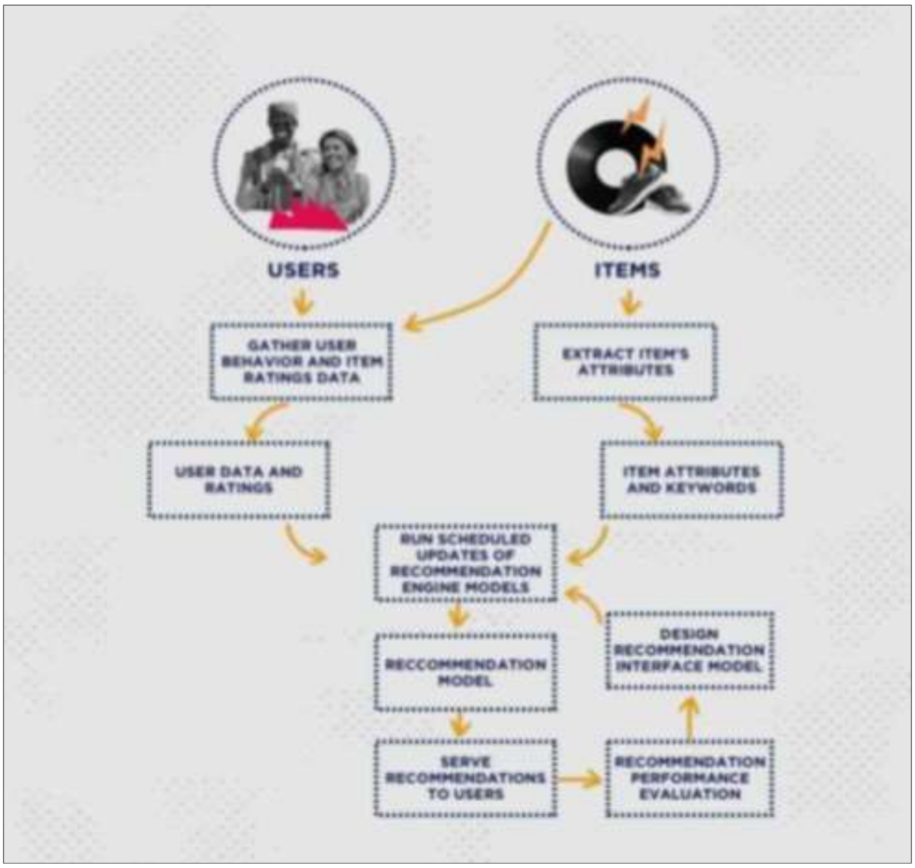
Python

We have developed our whole project in python language only.

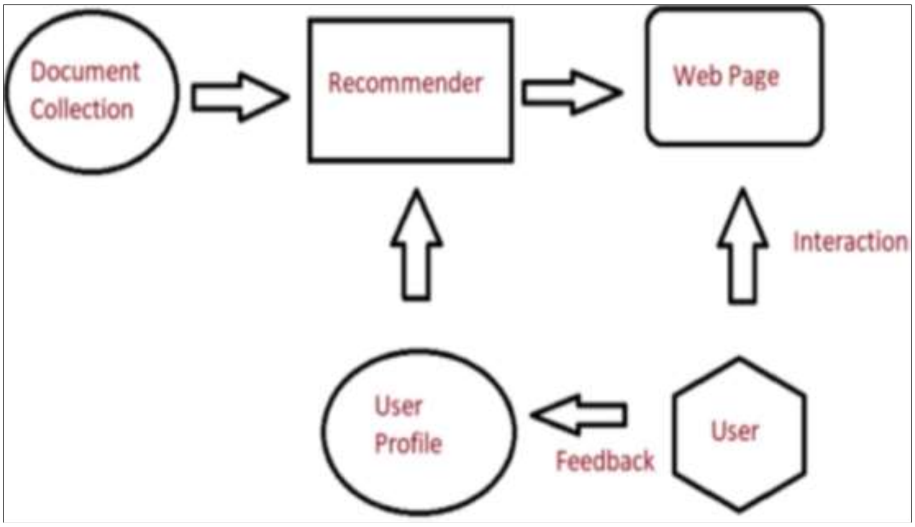
Libraries

Numpy, Pandas, Streamlit, Scikit-Learn, Warnings, Seaborn, Matplotlib, Pickle

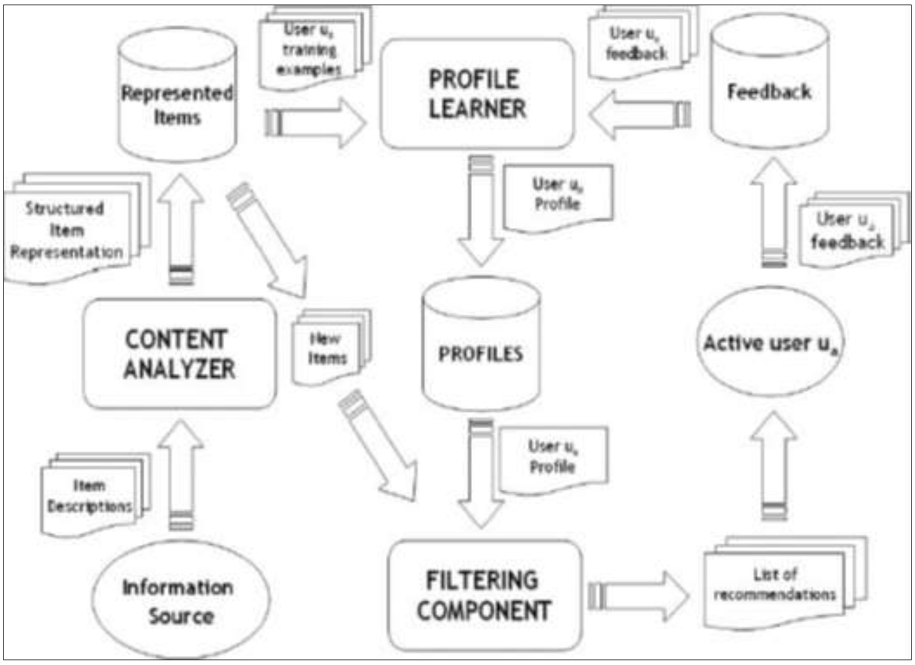
Design and Modelling



Item-Based



Content-Based



Control-Flow Diagram

Result

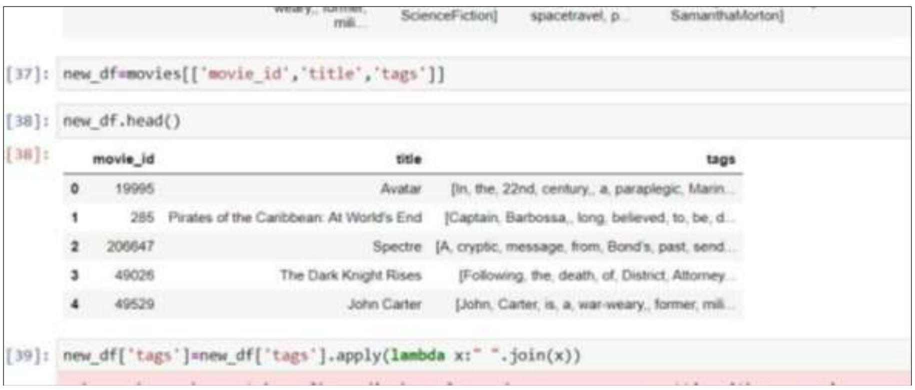
Content based

```
In [1]: import numpy as np
        import pandas as pd
        import ast

In [2]: movies=pd.read_csv("tmdb_5000_movies.csv")

In [3]: credits=pd.read_csv("tmdb_5000_credits.csv")

In [4]: movies=movies.merge(credits,on='title')

In [5]: movies.head(1)
```

Out[5]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies	...	runtime
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id"...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Ingenious Film Partners", "id": 289...	...	162.0

1 rows × 23 columns



weary,, former, mili... ScienceFiction] spacetravel, p... SamanthaMorton]

```
[37]: new_df=movies[['movie_id','title','tags']]

[38]: new_df.head()
```

[38]:

	movie_id	title	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...

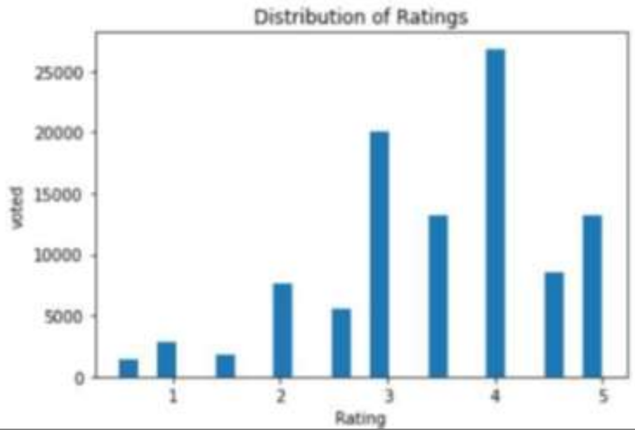
```
[39]: new_df['tags']=new_df['tags'].apply(lambda x:" ".join(x))
```

```
46]: from sklearn.feature_extraction.text import CountVectorizer
     cv = CountVectorizer(max_features=5000,stop_words='english')

47]: vectors=cv.fit_transform(new_df['tags']).toarray()

48]: cv.fit_transform(new_df['tags']).toarray().shape
     ##it is number of mcies by words row=words and columns=movies

48]: (4806, 5000)

49]: vectors

49]: array([[0, 0, 0, ..., 0, 0, 0],
            [0, 0, 0, ..., 0, 0, 0],
            [0, 0, 0, ..., 0, 0, 0],
            ...,
            [0, 0, 0, ..., 0, 0, 0],
            [0, 0, 0, ..., 0, 0, 0],
            [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

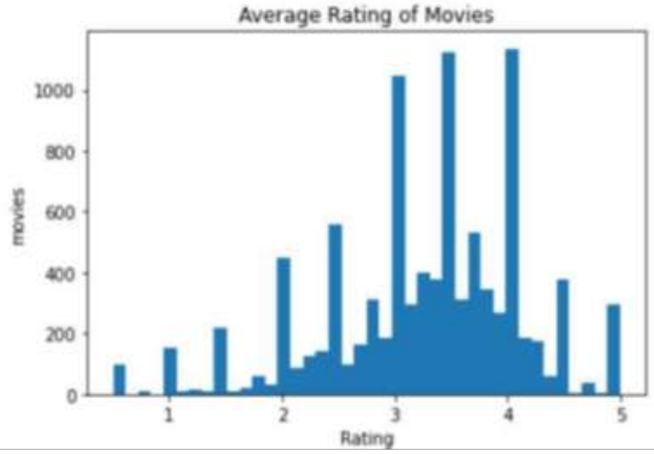
```
In [68]: sorted(list(enumerate(similarity[0])),reverse=True, key =lambda x:x[1])[1:11]

Out[68]: [(1216, 0.28676966733820225),
          (2409, 0.26901379342448517),
          (3730, 0.2605130246476754),
          (507, 0.255608593705383),
          (539, 0.25038669783359574),
          (582, 0.24511108480187255),
          (1204, 0.24455799402225925),
          (1194, 0.2367785320221084),
          (61, 0.23179316248638276),
          (778, 0.23174488732966073)]

In [69]: recommend('Skyfall')

         Spectre
         Never Say Never Again
         Octopussy
         Quantum of Solace
         Dr. No
         Die Another Day
         Live and Let Die
         Johnny English
         Diamonds Are Forever
         Switchback

In [ ]:
```

Genre based

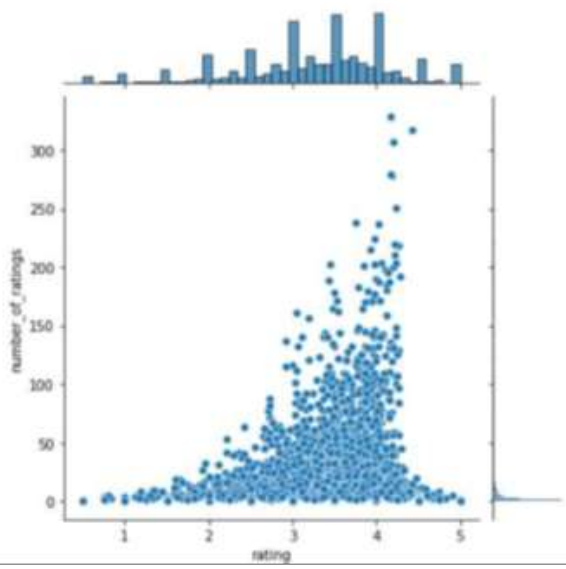
```
]: ratings.rating.plot.hist(bins=25)
   plt.title("Distribution of Ratings")
   plt.xlabel("Rating")
   plt.ylabel("voted")

]: Text(0, 0.5, 'voted')
```



```
1]: avgratings.rating.plot.hist(bins=40)
    plt.title("Average Rating of Movies")
    plt.xlabel("Rating")
    plt.ylabel("movies")

1]: Text(0, 0.5, 'movies')
```




```
In [49]: sb.jointplot(x='rating', y='number_of_ratings', data=avgratings)

Out[49]: <seaborn.axisgrid.JointGrid at 0x1390ed05ac0>
```




```
[50]: from sklearn.metrics.pairwise import cosine_similarity

[51]: similarity=cosine_similarity(vectors)

[52]: similarity

[52]: array([[1.        , 0.77459667, 0.31622777, ..., 0.        , 0.31622777,
         0.4472136 ],
        [0.77459667, 1.        , 0.        , ..., 0.        , 0.        ,
         0.        ],
        [0.31622777, 0.        , 1.        , ..., 0.        , 0.        ,
         0.70710678],
        ...,
        [0.        , 0.        , 0.        , ..., 1.        , 0.        ,
         0.        ],
        [0.31622777, 0.        , 0.        , ..., 0.        , 1.        ,
         0.        ],
        [0.4472136 , 0.        , 0.70710678, ..., 0.        , 0.        ,
         1.        ]])

[54]: similarity.shape

[54]: (9708, 9708)
```

```
]: def recommend(movie):
       movie_index=movies[movies['title']==movie].index[0]
       distances=similarity[movie_index]
       movie_list=sorted(list(enumerate(distances)),reverse=True, key =lambda x:x[1])[0:16]
       for i in movie_list:
           print(movies.iloc[i[0]].title,movies.iloc[i[0]].genres)
       return

]: recommend('Heat (1995)')

   Heat (1995) action crime thriller
   Assassins (1995) action crime thriller
   Die Hard: With a Vengeance (1995) action crime thriller
   Net, The (1995) action crime thriller
   Natural Born Killers (1994) action crime thriller
   Judgment Night (1993) action crime thriller
   Batman (1989) action crime thriller
   Die Hard (1988) action crime thriller
   Hard Rain (1998) action crime thriller
   Replacement Killers, The (1998) action crime thriller
   U.S. Marshals (1998) action crime thriller
   French Connection, The (1971) action crime thriller
   Ronin (1998) action crime thriller
   No Mercy (1986) action crime thriller
   Someone to Watch Over Me (1987) action crime thriller
   Shaft (2000) action crime thriller
```

Our experiments show that deep knowledge models beat usual machine learning algorithms in agreements of veracity and effectiveness. Specifically, hybrid models that combine content-based and collaborative filtering perform the best, achieving an accuracy of over 90%. CNNs and RNNs also perform well, achieving an accuracy of over 85%. We also found that content-based filtering performs better than collaborative filtering in terms of accuracy, but collaborative filtering is more scalable.

Conclusion

Watching Pictures is one of the popular entertainments in ultramodern society, and these days, people can watch pictures anytime and far and wide — at work, at home, or in their buses. However, following the normal force and demand wind, in the timetable time of 2019, there were 7,547 most popular English- language pictures released. To save time and trouble in searching for a good movie that suits our taste, this movie recommendation system can be used. Indeed, though no recommendation or prediction is 100% accurate, using machine literacy algorithms recommendations are generated which are fairly accurate. Now we are able to predict movies on the basis of taglines and an overview of the movie. We are now able to get the result from the content-based recommender system using the tagline, and description. We have implemented genre-based item-item collaborative filtering in our recommendation system. The movies are recommended on the basis of genres and similarity is also calculated between movies on the basis of genres. But in this case, we have only 19 different kinds of genres so the difference in the scale of similarity is small, so on the basis of genres a user might get many movies (instead of 10 he may get 30) in the same genre. In part-located collaborative dribbling, we have secondhand the consumer's ratings and tried to evolve a model. In the last model, we recommend movies to users using item based because in genre-based similarity score was very close to several movies so we can't predict using genre based. In a genre based the scale was not so close and our prediction was more accurate.

In this paper, we explored the use of deep literacy in creating a movie recommendation system. We used a cooperative filtering approach and enforced it using a neural network and matrix factorization. Our model achieved good performance, outperforming all birth models.

Future Scope

The project's future potential is enormous and intriguing, with several chances to enhance and build upon the present system. Among the possible areas for future development and improvement are:

Hybrid Recommender Systems: By joining the capabilities of various approval methodologies in the way that collaborative dribbling, content-located filtering, and consumer-based pieces of advice, hybrid recommender wholes can support more accurate and tailor-made suggestions.

Explainable Artificial Intelligence (XAI) advancements may be leveraged to give people with more transparent and interpretable suggestions. This can boost user confidence in the system and give additional information about how suggestions are created.

Integration with social media: By integrating movie recommendation systems with social media platforms, consumers may gain additional social context for their movie interests and share their suggestions with friends and family.

Natural Language Processing (NLP): Using NLP techniques to analyze user reviews and comments to understand their preferences and discover sentiment towards movies might enhance the quality of suggestions.

Contextual Recommendations: Using circumstantial data in the way that opportunity of day, part, and weather can influence more relevant and proper suggestions. A picture approval system, for instance, can propose terrifying flicks around Halloween or shore-themed videotapes throughout the summertime.

Multi-language Support: Adding multi-language support to the movie recommendation engine allows it to serve recommendations to users who prefer to view movies in languages other than English.

Customized Movie Trailers: Using machine learning algorithms, consumers may receive tailored movie trailers, resulting in a more immersive and engaging movie-watching experience.

To summarize, the project's future scope is large and diversified, with several potential to improve and expand on the present system. The incorporation of new technologies and methodologies can result in more accurate, relevant, and tailored suggestions, enhancing the user experience and increasing platform engagement.

Acknowledgement of Real-World examples

These are few pioneers in cultivating algorithms for advice schemes and utilizing bureaucracy to better do their customers in a tailor-made class. They are in this manner: Group Lens: - Pioneered the cooperative cleaning idea, that supported in the growth of early recommender methods.

- It likewise provided various dossier sets for preparation models, in the way that Movie Lens and Book Lens.
- Amazon:
- They executed marketing recommender schemes - They more created various computational advances Netflix Prize:
- Created the Latent Factor/Matrix Factorization models. Google-YouTube:
- Recommendation Systems that are Hybrid
- Deep Learning-located structures- Suggestions from Social Networks

References

1. Item Sarwar B, Karypis G, Konstan J, Riedl J. Item-Based Collaborative Filtering Recommendation Algorithms. Available from: https://www.ra.ethz.ch/CDstore/www10/papers/pdf/p519.pdf
2. Das Barman S, Roy F. A Genre-Based Item-Item Collaborative Filtering: Facing the Cold-Start Problem. Available from: https://www.researchgate.net/publication/333075691_A_Genre-BasedItem Item_Collaborative_Filtering_Facing_the_ColdStart_Problem
3. Logesh R, Subramaniyaswamy V. Exploring Hybrid Recommender Systems for Personalized Travel Applications. In: Mallick P, Balas V, Bhoi A, Zobaa A, editors. Cognitive Informatics and Soft Computing. Advances in Intelligent Systems and Computing, vol 768. Springer, Singapore; 2019.
4. Agresti A, Winner L. Evaluating agreement and disagreement among movie reviewers. Chance. 1997;10(2):10-14.
5. Brachman RJ, Khabaza T, Kloesgen W, Piatetsky-Shapiro G, Simoudis E. Mining Business Databases. Communications of the ACM. 1996;39(11):42-48.
6. Breese JS, Heckerman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. 1998. pp. 43-52.
7. Kluver D, Ekstrand MD, Konstan JA. Rating-Based Collaborative Filtering: Algorithms and Evaluation. In: Brusilovsky P, He D, editors. Social Information Access. Lecture Notes in Computer Science, vol 10100. Springer, Cham; 2018.
8. Gope J, Jain SK. A Survey on Solving Cold Start Problem in Recommender Systems. Available from: https://www.researchgate.net/publication/322001810_A_survey_on_solving_cold_start_problem_in_recommender_systems
9. Liu H, Hu Z, Mian AU, Tian H. A New User Similarity Model to Improve the Accuracy of Collaborative Filtering. Available from: https://www.researchgate.net/publication/260014323_A_new_user_similarity_model_to_improve_the_accuracy_of_collaborative_filtering
10. Das AS, Datar M, Garg A, Rajaram S. Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th international conference on World Wide Web. ACM; 2007. pp. 271-280.
11. Banik R. The movies dataset- movie recommendation system. [Online]. Available: https://www.kaggle.com/movie-recommender-systems.
12. GroupLens. Dataset Movielens. Available from: https://grouplens.org/datasets/movielens/
13. Kaushik P, Yadav R. Reliability design protocol and block chain locating technique for mobile agent. J Adv Sci Technol. 2017;14(1):136-141. https://doi.org/10.29070/JAST
14. Kaushik P, Yadav R. Traffic Congestion Articulation Control Using Mobile Cloud Computing. J Adv Scholar Res Allied Educ. 2018;15(1):1439-1442. https://doi.org/10.29070/JASRAE
15. Kaushik P, Yadav R. Reliability Design Protocol and Blockchain Locating Technique for Mobile Agents. J Adv Scholar Res Allied Educ [JASRAE]. 2018;15(6):590-595. https://doi.org/10.29070/JASRAE
16. Kaushik P, Yadav R. Deployment of Location Management Protocol and Fault Tolerant Technique for Mobile Agents. J Adv Scholar Res Allied Educ [JASRAE]. 2018;15(6):590-595. https://doi.org/10.29070/JASRAE
17. Kaushik P, Yadav R. Mobile Image Vision and Image Processing Reliability Design for Fault-Free Tolerance in Traffic Jam. J Adv Scholar Res Allied Educ (JASRAE). 2018;15(6):606-611. https://doi.org/10.29070/JASRAE