

MOVIE RECOMMENDATION SYSTEM

Student ID – 780556

Student Name – Chankya Patel

Supervisor Name – Savita Seharawat

Table of Contents

| | |
|--------------------------------------|-----------|
| Abstract..... | 3 |
| Keywords..... | 3 |
| Tools..... | 4 |
| GitHub..... | 4 |
| Introduction..... | 4 |
| Content Based Filtering | 5 |
| Collaborative Filtering..... | 5 |
| Hybrid Filtering..... | 5 |
| Literature Review | 6 |
| Methodology:..... | 10 |
| Extracting Data | 11 |
| Exploratory Analysis | 11 |
| Dataset (TMDB): | 11 |
| Statistical Data..... | 12 |
| Data pre-processing | 12 |
| Null Values | 12 |
| AST | 13 |
| Visualization | 14 |
| Soup..... | 15 |
| Finding Similarities..... | 15 |
| CountVectorizer | 15 |
| Cosine_similarity | 16 |
| Recommendation Function | 16 |
| Name..... | 17 |
| Conclusion | 17 |
| References..... | 18 |

Abstract

A recommendation system is simple algorithm which is aimed to provide the relevant information to a user by discovering patterns in a dataset. To simply put, it will give the user relevant information based on their history data. For instance, we can take Netflix, Amazon, Spotify recommendation system how they recommend things to a user. They have the user's history for watching a movie or buying some things from their website. After watching some movies, you will start getting the recommended movies which are similar to that movie. The recommendation engine makes suggestions by learning and understanding the patterns in you watch history and then applies those patterns and findings to make new suggestions.

After searching a lot on internet and referring a lot of research papers, I got to know that there are lots of recommendations systems using different techniques. Here, I will use content-based filtering to make the recommendation system.

Purpose of choosing this project is that today, everyone wants an intelligent streaming platform that can understand their preferences and tastes without merely running on autopilot. Netflix Recommendation Engine (NRE) is the most successful algorithm of Netflix, and it is the core of their Netflix product. Recommending the right things to a user keeps their interest of using your product.

Keywords

- Machine Learning, Recommendation System, Content-based, text to vector, cosine similarity, TMDB

Tools

- Jupyter, requests, pandas, numpy, cosine_similarity, CountVectorizer, linear_kernel, TfidfVectorizer, json, KNNImputer, AST, itertools, powerbiclient, report.

GitHub

- https://github.com/chankyapatel/Movie_Recommendation_System

Introduction

In today's world there are lots of information on internet from which to consume and what not to consume is really difficult. Even on YouTube, when you want to watch a video of particular concepts, generally, there are lot of videos available out there for you. But one question comes here, what if the results were not ranked appropriately? Well, in that case, we would probability spend more time to find the best possible video which suits us and satisfies our need. This recommendation results are when you search something on a website. So, basically the job of recommender system is to suggest the most relevant items to the user. Recommendation systems are used in YouTube, Amazon, Netflix, Amazon prime for movie recommendations and so on. Whatever you do on such websites, there is a system which sees your behaviour and then ultimately suggest things/items with which you are highly likely to engage. This system is called recommendation system. Website like Amazon, Netflix etc. use movie recommendation to increase their revenue of profit by ultimately improving the user experience.

We have a lot of data available at our exposure and we need to filter the data in order to consume it because generally we are not interested in each and everything available to us. In order to filter the data, we need some filtering techniques. There are different types of filtering techniques or movie recommendation algorithm over which a recommendation system can be based upon.

Major filtering techniques or movie recommendation algorithms are as follows:

- Content Based Filtering
- Collaborative Filtering
- Hybrid Filtering

Content Based Filtering

- Content based systems are used in a wide variety of domains, such as web pages, product description, news, music features, and so on. In most cases features are extracted from these various sources to convert them into a keyword-based vector-space representation. A content-based model is specific to a given user. Therefore, a user-specific model is constructed user interests in items, based on their history of either buying or rating items.

Collaborative Filtering

- Collaborative methods for recommender systems are methods that are based solely on the interactions recorded between users and items to produce new recommendations.

Hybrid Filtering

- A hybrid recommendation system is special type of recommendation system which can be considered as the combination of content and collaborative filtering method. When recommendation system based on content based and collaborative face the issue when there is not enough data to learn the relationship between user and items. In these kind of cases, the third type of approach is used to build recommendation system named as Hybrid Filtering.

Literature Review

Sang-Min Choi, et. al. (2012): 8079-8085 mentioned about the shortcomings of collaborative filtering approach and to avoid this problem the authors decided to use category information because the category information is present for the newly created content. The reason for this is even if the new content does not have enough ratings or enough views, still it will come up in recommendation list. How it will come up in recommendation – with the help of category and genre information. Therefore, even when a new movie comes it can be recommended by the recommendation system.

The solution of using the hybrid approach of movie recommendation is proposed by George Lekakos, et. al. (2008): 55-70. Therefore, the authors have come up with a hybrid approach of recommendation system which includes both content-based filtering and collaborative filtering. The solution is already implemented in ‘MoRe’, it is a movie recommendation system. The authors have not used Pearson correlation for the sake of pure collaborative filtering. Instead of using Pearson correlation for content-based recommendation system, they decided to use cosine similarity by taking into consideration movie director, cast, producers and the movie genre. This approach works on collaborative filtering and show recommendation based on content-based filtering when certain criterion is met. The authors have used collaborative filtering technique as their main approach.

Debashis Das, et. al. (2017) wrote about the different types of recommendation system and their general information on the survey paper of recommendation system. He has also mentioned about personalized recommendation system also non personalized system. He has explained the user based collaborative filtering and item based collaborative filtering. He has also mentioned about the merits and demerits of different recommendation system.

Movie Recommendation System

There is an approach called 'Weighted KM-Slope-VU' which is based on collaborative filtering and that was proposed by Jiang Zhang, et. al. (2019): 180-191. He has divided the users into clusters of similar users with the help of k-means clustering. Later, they selected a virtual opinion leader from each cluster which represents the all the users in that particular cluster. Instead of processing complete user-item rating matrix, the authors processed virtual opinion leader-item matrix which is of small size. By this approach the time for getting recommendation gets less.

S. Rajarajeswari, et. al. (2019). 329-340. discussed about the simple recommender system, Content based recommender system & collaborative filtering-based recommender finally came with the solution consisting of Hybrid Recommendation System. The authors have taken decision to use cosine similarity. Their system got 30 movie recommendations using cosine similarity after that they implemented the recommender system by filtering movies based on user ratings. The system takes only the recent movie which the user has watched because their recommendation system takes only one movie as input.

Muyeed Ahmed, et. al. (2018) came up with a solution using k-means clustering algorithm. Authors have separated similar users by using cluster. Later, authors have created a neural network for each cluster for recommendation purpose. The proposed system consists of steps like Data Preprocessing, Principal Component Analysis, clustering, Data Processing for Neural Network, and Building Neural Network, User rating, user preference, and user consumption ratio have been taken into consideration. After clustering phase, for the purpose of predicting the ratings which the user might give to the unwatched movies, the authors have used neural network. Finally, recommendations are made with the help of predicted high ratings.

Gaurav Arora, et. al. (2014): 765-770 have proposed a solution of movie recommendation which is based on users' similarity. The research paper is very general in the sense that the authors

have not mentioned the internal working details. In the Methodology section, the authors have mentioned about City Block Distance and Euclidean Distance but have not mentioned anything about cosine similarity or other techniques. The authors stated that the recommendation system is based on hybrid approach using context based filtering and collaborative filtering but neither they have stated about the parameters used, nor they have stated about the internal working details

V. Subramaniaswamy, et. al. (2017): 54-63 have proposed a solution of personalized movie recommendation which uses collaborative filtering technique. Euclidean distance metric has been used in order to find out the most similar user. The user with least value of Euclidean distance is found. Finally, movie recommendation is based on what that particular user has best rated. The authors have even claimed that the recommendations are varied as per the time so that the system performs better with the changing taste of the user with time.

Harper, et. al. (2015): 1-19 mentioned the details about the Movie Lens Dataset in their research paper. This dataset is widely used especially for movie recommendation purpose. There are different versions of dataset available like Movie Lens 100K / 1M / 10M / 20M / 25M / 1B Dataset. The dataset consists of features like user id, item id / movie id, rating, timestamp, movie title, IMDb URL, release date, etc. along with the movie genre information

According to R. Lavanya, et. al. (2021), pp. 532-536, in order to tackle the information explosion problem, recommendation systems are helpful. Authors mentioned about the problems of data sparsity, cold start problem, scalability, etc. Authors have done a literature review of nearly 15 research papers related to movie recommendation system. After reviewing all these papers, they observed that most of the authors have used collaborative filtering rather than content-based filtering. Also, the authors noticed that a lot of authors have used hybrid-based approach.

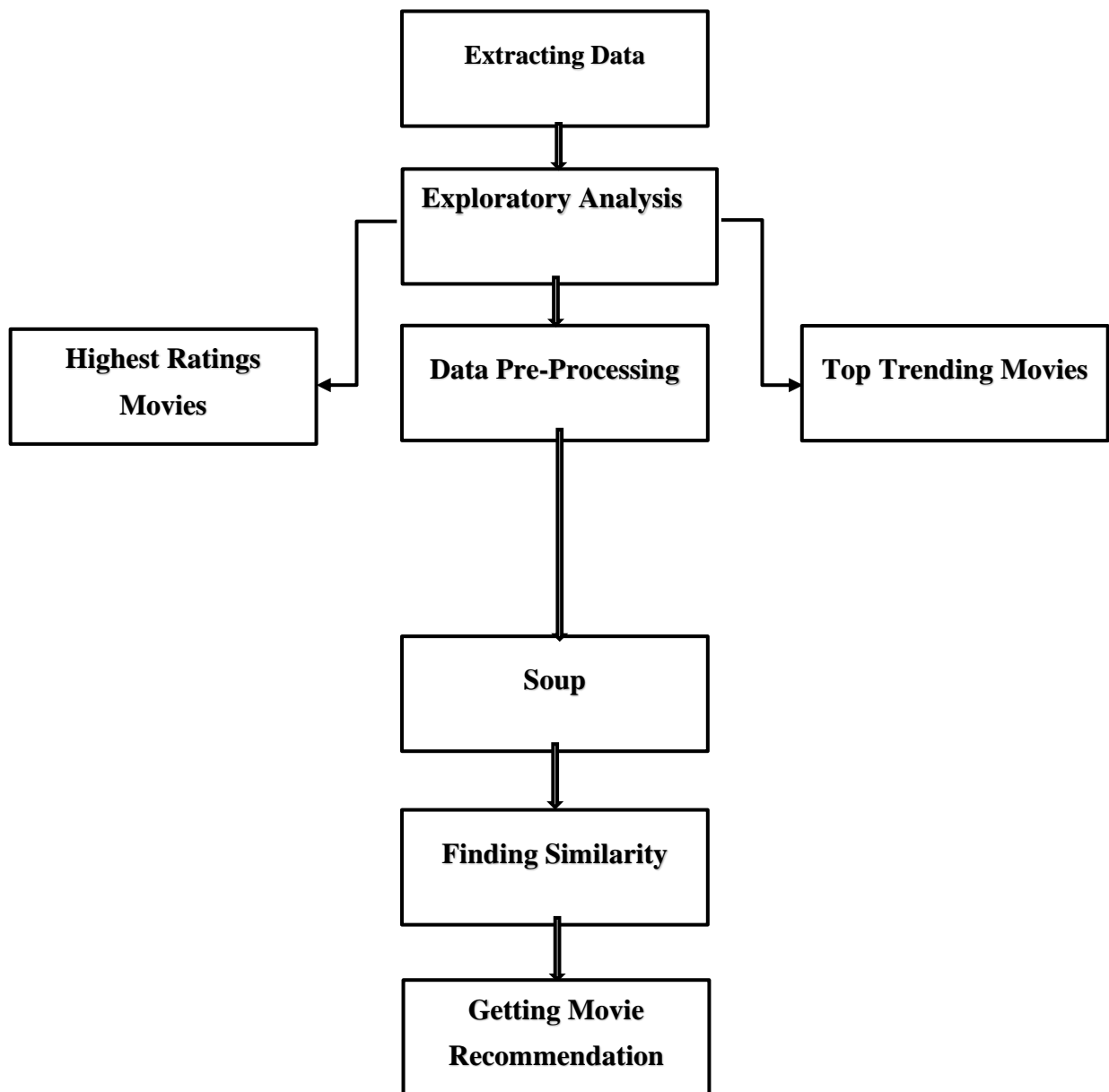
Movie Recommendation System

Even though a lot of research has been done on recommendation systems, there is always a scope for doing more in order to solve the existing drawbacks.

Ms. Neeharika Immaneni, et. al. (2017), pp. 2193-2200 proposed a hybrid recommendation technique which takes into consideration both content-based filtering approach as well as collaborative filtering approach in a hierarchical manner in order to show a personalized movie recommendation to the users. The most unique thing about this research work is that the authors have made movie recommendations using a proper sequence of images which actually describe the movie story plot. This actually helps for better visuals. The author have also described the graph based recommendation system, content-based approaches, hybrid recommender systems, collaborative filtering systems, genre correlations based recommender system, etc.

Md. Akter Hossain, et. al. (2018), pp. 443-448 proposed NERS which is an acronym for neural engine-based recommender system. The authors have done interaction between 2 datasets carefully. Moreover, the authors stated that the results of their system are better than the existing systems because they have incorporated the usage of general dataset as well as the behaviour-based dataset in their system.

Methodology:



Extracting Data

Created a developer account on TMDB website. Provided the description on how I am going to use the data and made request for API key. After some days they approved my request and gave me the API key.

Exploratory Analysis

Dataset (TMDB):

Table 1: Data Decsription

| Features | Data Type | Description |
|----------------------|-----------|-------------------------------------------|
| budget | Float | It contains the budget of the movie |
| genre | Object | Different types of genres of each movie |
| Id | Float | Id of the movie |
| original_language | Object | Original language that movie is made in |
| original_title | Object | Title of the movie |
| Overview | Object | Brief of movie description |
| popularity | Float | Counting the number of ratings |
| production_companies | object | Production companies of the movie |
| release_date | Datetime | Date when movie was released |
| revenue | Float | How much the movie gain |
| runtime | Float | How long a movie is (min) |
| spoken_languages | object | Different languages a movie is dubbed in |
| title | Object | Title of the movie |
| Ratings | Float | How much ratings a movie has (4.0 – 10.0) |
| vote_count | float | Total count of votes |

Statistical Data

Table 2: Statistical Data

| | budget | id | popularity | revenue | runtime | Ratings | vote_count |
|-------|----------|----------|------------|----------|----------|----------|------------|
| Count | 779 | 779 | 779 | 779 | 779 | 779 | 779 |
| Mean | 23939597 | 494.6534 | 22.54806 | 1.21E+08 | 114.2054 | 7.215661 | 3209.741 |
| Std | 40949454 | 274.4348 | 39.14602 | 2.11E+08 | 28.85557 | 0.743687 | 4374.002 |
| Min | 0 | 11 | 0.6 | 0 | 1 | 4 | 1 |
| 25% | 80000 | 246.5 | 8.2435 | 3839.5 | 98 | 6.8 | 394.5 |
| 50% | 6000000 | 507 | 13.004 | 28350000 | 111 | 7.3 | 1403 |
| 75% | 28000000 | 750.5 | 22.197 | 1.48E+08 | 127 | 7.7 | 4223.5 |
| max | 3E+08 | 949 | 593.963 | 2.19E+09 | 233 | 10 | 27583 |

Data pre-processing

Null Values

Table 3: Null Values

| Features | No.160 of Null values |
|----------|-----------------------|
| Budget | 160 |
| Genres | 160 |
| Homepage | 689 |
| Id | 160 |

| Features | No.160 of Null values |
|----------------------|-----------------------|
| original_language | 160 |
| original_title | 160 |
| Overview | 160 |
| Popularity | 160 |
| production_companies | 160 |
| production_countries | 160 |
| release_date | 160 |
| revenue | 160 |
| Runtime | 160 |
| spoken_languages | 160 |
| Title | 160 |
| Ratings | 160 |
| vote_count | 160 |

Initially, decided to replace them but after checking them in the excel got to know that all these null values extracting 'homepage' is just whole empty row, which does not add any meaning to the data. Therefore, decided to drop the whole row.

AST

To transform the data in a right format which is easier to read I have used AST (Abstract Syntax Tree) library in python. Created function for that and applied it on features such as: genres, production_countries, spoken_languages.

Visualization

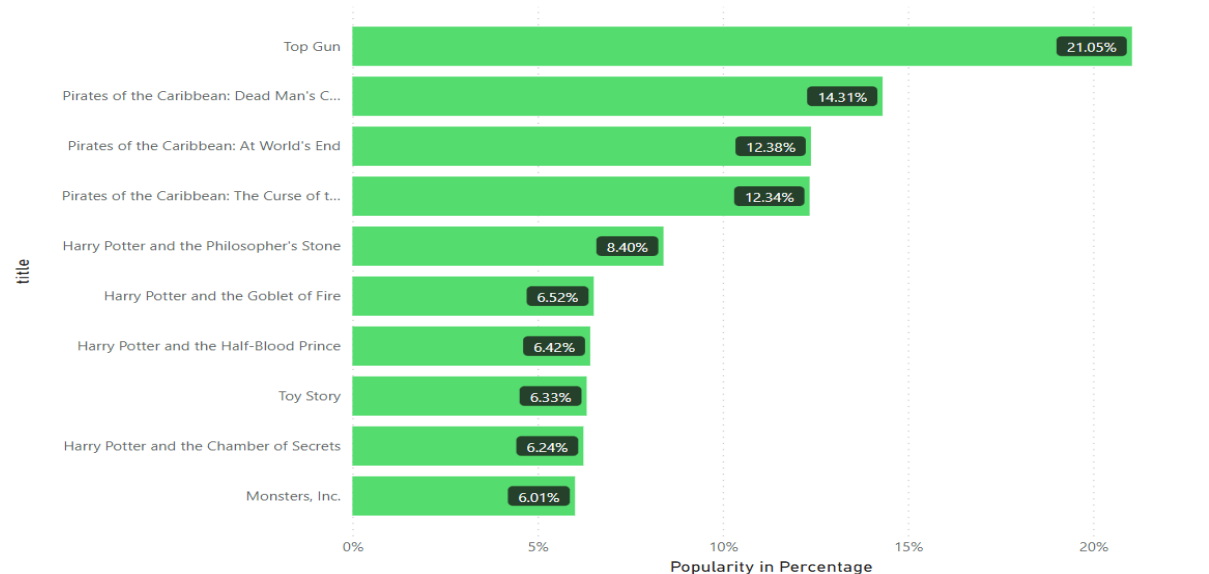


Figure 1: To Trending Movies

For visualizing the 'Top Trending Movies' powerbiclient tool has been used. powerbiclient tool is library in python which helps user to call the report from Power BI. It is in form of percentage because it was difficult to explain with the count of popularity, therefore, created this report with percentage.

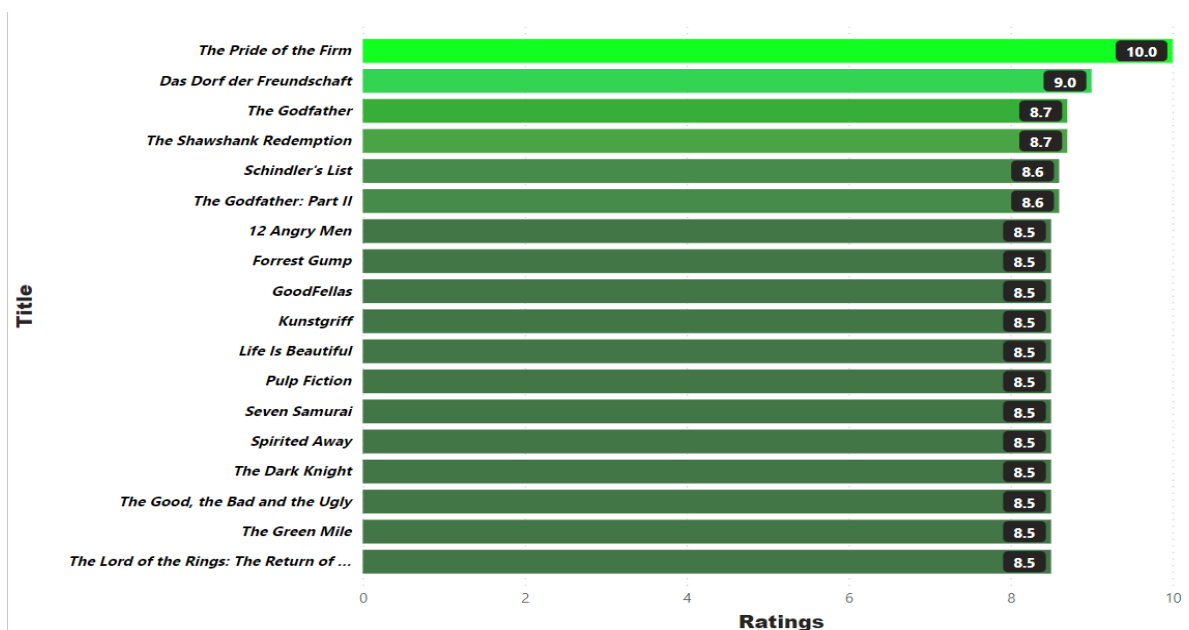


Figure 2: Top Rated Movies

For visualizing the ‘Top Rated Movies’ same library has been used. Called the report from powerbi to jupyter. Used ‘title’ and ‘Ratings of the movies. All the numbers represent the ratings of that particular movies.

Soup

Basically, SOUP is used to join multiple features from the dataset. This is one of the core functions of Recommendation system. This function will join all the features that we put, and it combines all the data from those and makes it a one sentence. For Instance, joining several functions such as: ‘spoken_languages’, ‘genres’, ‘overview’ and ‘production_countries’.

Here is the output:

```
english adventure action sciencefiction p r i ...  
english animation family n e m o , a n a d v e...  
english comedy drama romance a m a n w i t h a...
```

Figure 3: Soup

Finding Similarities

CountVectorizer

CountVectorizer is used to transform a given text into a vector on the basis of frequency of the word that occurs in the entire text. For that created a vector containing countvectorizer. Then applied it in the SOUP.

Cosine_similarity

Cosine_similarity measures the similarity between vector lists by calculating the cosine angle between the two-vector list. Then put this function in those two vectors which were created early in countvectorizer.

Recommendation Function

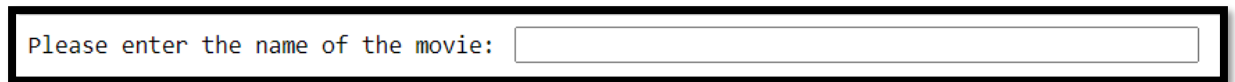
Created the function named (get_recommendation) which contains 'title', 'cosine_similarity'

Then I have created three vectors.

- First vector will make tuples.
- Second vector is the core of this function.
 - o It contains sorted (similarity score), lambda function (key=lambda x:x[1])
 - o sorted (similarity score) will sort the list of tuples in descending order based on similarity score.
 - o Lambda function will create an inline function
 - o x where, x is taken as an input by inline function and it will return x[1]
 - o x[1] it is the second element of x.
- Third vector contains similarity score[1:11]
 - o Similarity score[1:11] here, took 1 because indexing in python starts from 0.

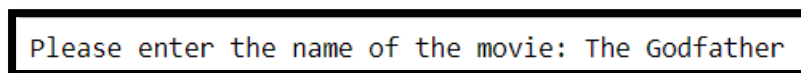
Name

Created a string in which user can put a movie name and after running the `get_recommendation` function it gives recommended movies for the input movies name.



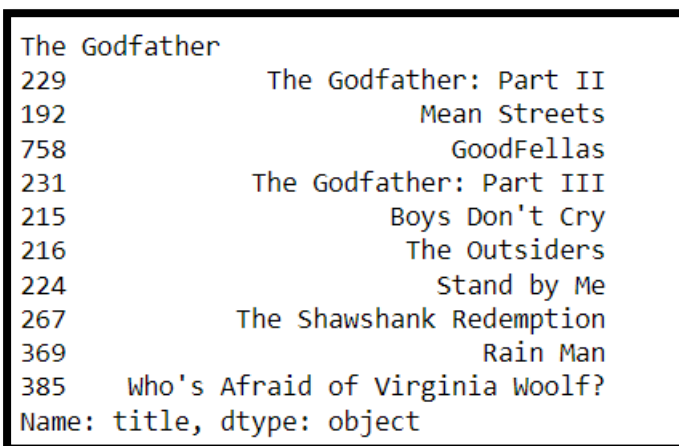
```
Please enter the name of the movie: 
```

Figure 4: String to enter movie name



```
Please enter the name of the movie: The Godfather
```

Figure 5: The Godfather



```
The Godfather
229          The Godfather: Part II
192                      Mean Streets
758                      GoodFellas
231          The Godfather: Part III
215                      Boys Don't Cry
216                      The Outsiders
224                      Stand by Me
267          The Shawshank Redemption
369                      Rain Man
385  Who's Afraid of Virginia Woolf?
Name: title, dtype: object
```

Figure 6: Recommended Movies

Conclusion

Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. These systems helps users find items they want to buy or watch from a business. Recommender systems benefits users by enabling them to find items they like.

References

- [1].Choi, Sang-Min, Sang-Ki, K., & Yo-Sub, H. (2012). *"A movie recommendation algorithm based on genre correlations"*. Republic of Korea: Expert Systems with Applications. 8079-8085
- [2].Das, Debashis, Laxman Sahoo, & Sujoy Datta. (2017). *"A survey on recommendation system"*. NY, USA: International Journal of Computer Applications.
- [3].Lekakos, George, & Petros Caravelas. (2008). *"A hybrid approach for movie recommendation"*. Multimedia tools and applications 36.1.
- [4].Zhang, Jiang, et al. "Personalized real-time movie recommendation system: Practical prototype and evaluation." Tsinghua Science and Technology 25.2 (2019): 180-191
- [5].Rajarajeswari, S., et al. "Movie Recommendation System." Emerging Research in Computing, Information, Communication and Applications. Springer, Singapore, 2019. 329-340
- [6].Ahmed, Muyeed, Mir Tahsin Imtiaz, and Raiyan Khan. "Movie recommendation system using clustering and pattern recognition network." 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2018
- [7].Arora, Gaurav, et al. "Movie recommendation system based on users' similarity." International Journal of Computer Science and Mobile Computing 3.4 (2014): 765-770
- [8].Subramaniaswamy, V., et al. "A personalised movie recommendation system based on collaborative filtering." International Journal of High Performance Computing and Networking 10.1-2 (2017): 54-63

- [9].Harper, F. Maxwell, and Joseph A. Konstan. "The movielens datasets: History and context." *Acm transactions on interactive intelligent systems (tiis)* 5.4 (2015): 1-19.
- [10]. R. Lavanya, U. Singh and V. Tyagi, "A Comprehensive Survey on Movie Recommendation Systems," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 532-536, doi: 10.1109/ICAIS50930.2021.9395759
- [11]. N. Immaneni, I. Padmanaban, B. Ramasubramanian and R. Sridhar, "A meta-level hybridization approach to personalized movie recommendation," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 2193-2200, doi: 10.1109/ICACCI.2017.8126171
- [12]. M. A. Hossain and M. N. Uddin, "A Neural Engine for Movie Recommendation System," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEICT), 2018, pp. 443-448, doi:10.1109/CEEICT.2018.8628128