WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Movie Recommendation System

Rishabh Sharma, Saaransh Mahipal, Samar Srivastav, Shubhangi Singh, Krishan Kumar

Dept. of Information Technology, Inderprastha Engineering College

ABSTRACT

Today, recommender systems have changed the way we search for what we are interested in. The most popular areas of interest for the recommender system are books, news, articles, music, videos and movies. This document proposes a movie recommendation system. An approach that analyzes information provided by collaborative filtering-based and content-based recommends the movie that is most suitable for the user at that time. The list of recommended movies is sorted by the ratings that these movies received from previous users, using various algorithm for this purpose. User also helps you find featured movie in an efficient and effective way, based on the movie experience of other users, without wasting a lot of time on wasted browsing. This system was developed using Python and a PostgreSQL server. The presented recommender system uses different types of knowledge and data about users, available items, and past transactions stored in a customized database to generate recommendations. This allowsusers to easily browse the recommendations and findthe movie of their choice.

Keywords: Recommender, Recommendation System, Machine Learning, Content, Content Based, Data Filtering, Collaborative, Collaborative Filtering,

Introduction:

The system's major goal is to give the greatest possible user experience. As a result, businesses attempt to link users with the most relevant information based on their previous behaviour and to keep them hooked on their content.

The recommender system suggests which text to read next, which movie to view, and which product to buy, so increasing the stickiness of any product or service. Its proprietary algorithms are designed to forecast a user's interest, recommend various items to them in a variety of ways, and maintain that attention until the conclusion.

We see this system in action on a daily basis, it goes without saying. Many online merchants employ machine learning to generate sales through recommender systems (ML).

Collaborative filtering and content-based filtering (also known as the personality-based approach) are commonly used in recommender systems, as are other systems such as knowledge-based systems. Collaborative filtering techniques create a model based on a user's previous behaviour (things purchased or picked in the past and/or numerical ratings provided to those items) as well as comparable decisions made by other users. This model is then used to estimate which things (or item ratings) the user would be interested in. Filtering methods based on contentTo recommend further items with similar attributes, use a sequence of distinct, pre-tagged characteristics of an item. Current recommender systems are often hybrid systems that combine one or more methodologies. By contrasting two early music recommender systems – Last.fm and Pandora Radio – the distinctions between collaborative and content-based filtering may be seen. • Last.fm builds a "station" of recommended music by analyzing what bands and individual tracks a user listens to on a regular basis and comparing them to other users' listening habits. Last.fm will play tunes that you don't like, but that are frequently played by other people with similar interests. Because this method is based on behaviour This is a sample of what usersmight expect. collaborative filtering technology. • Pandora uses the properties of the song or artist (400 attributes provided by the Music Genome Project) A "station" that plays music with similar characteristics. User feedback Used to improve station results and not emphasize specific results Attributes when the user "dislikes" a particular song Highlights other When a user enjoys a music, they can add attributes to it. This is an example of a content-driven strategy. Each system type has its own set of advantages and disadvantages. one of them Last.fm, for example, demands a great deal of data from people who make correct suggestions. This is an example of a common cold start problem. Filtering system that is coordinated. Pandora, on the other hand, requires There is little information at first, and the scope is significantly narrower. There may only be similar recommendations. (Seed of origin). Adopting and implementing this approach on their websites has resulted in a significant number of sales for several retail companies. Netflix and Amazon, two forerunners in the use of recommenders, have released their algorithms for recommendation systems to hook their users.

Before getting into the details, it's important to understand that this approach eliminates irrelevant and duplicated data. Before it is shown to the front users, it intelligently filters out any information. As we want more personalised content delivered to our daily feeds, the recommender system has become a hot topic. I suppose we're all aware with YouTube's suggested videos, and we've all been the victims of late-night Netflix binge viewing.

Method

There are two primary types of recommendation systems, each with different sub-types. Depending on goals, audience, the platform, and what you're recommending, these different approaches can be employed individually, though generally, the best results come from using them in combination.

1.Content-Based

2.Collaborative filtering

3.Hybrid approach

1.Content Based:

The data generated by a user is used by the Content-Based Recommender System. Data can be generated either intentionally (by clicking likes) or intuitively (by watching videos). This information will be used to generate a user profile for the user, which will include the metadata associated with the things with which the user interacted. The more information it receives, the more precise the system or engine becomes. The data generated by a user is used by the Content-Based Recommender System. Data can be generated either intentionally (by clicking likes) or intuitively (by watching videos) (like clicking on links). This information will be used to generate a user profile for the user, which will include the metadata associated with the things with which the user interacted. The more information it receives, the more precise the system or engine becomes.

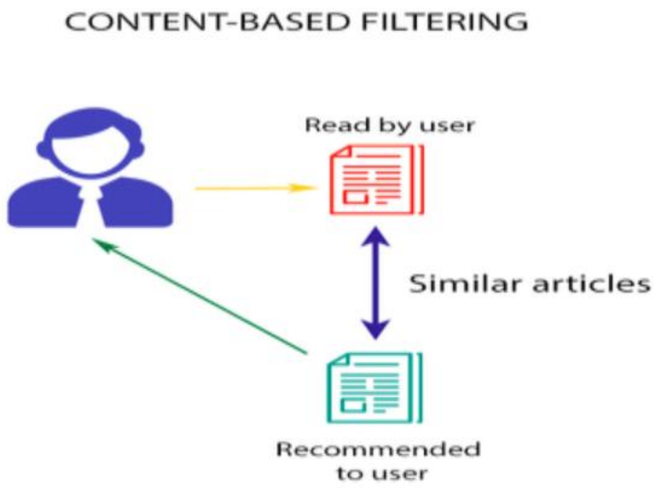

Figure 1 Content Based Approach.

These two tactics can be used in a variety of ways. To begin, the user is given a list of features from which to choose the most interesting ones.Second, the algorithms maintain track of all the products that the user has previously selected, forming the customer's behavioral data. The buyer's profile shapes the buyer's rating by rotating around the buyer's choices, tastes, and preferences. It shows how many times a single consumer clicks on things that piqued his or her attention, as well as how many times such products were liked in wish lists.

A likeness between the items is used in content-based filtering. The product's closeness and similarity are determined by the item's related content. When we talk about content, we're talking about the genre and the object.

The following are some of the more specific forms of content-based recommendation systems:

By Content Similarity: This method, which is the most basic sort of content-based recommendation system, involves recommending content that is similar based on its metadata. This method is appropriate for catalogues with a lot of rich metadata and little traffic in relation to the quantity of products in the catalogue.

By Popular Content Promotion: This entails emphasising product suggestions based on the product's inherent characteristics that may appeal to a broad audience: price, feature, popularity, and so on. This technique can also take into account the information's freshness or age, allowing the most current content to be used for suggestions. This is frequently utilised when the majority of the content is new.

2. Collaborative filtering

The technique of filtering or evaluating objects utilising the opinions of others is known as collaborative filtering. While the phrase collaborative filtering (CF) has only been around for a few years, it is based on something that humans have done for centuries: sharing thoughts with others.

Figure 2Collaborative filtering approach

Collaborative filtering is a widely used method to the creation of recommender systems. Collaborative filtering is founded on the idea that people who have agreed in the past will agree again in the future, and that they will enjoy comparable products. The technology creates suggestions based solely on rating profiles for various persons or things. They generate recommendations utilising this neighbourhood by seeking peer users/items with a rating history similar to the current user or item. Memory-based and model-based collaborative filtering approaches are the two types.

By User Similarity: This method entails grouping users based on their actions and making recommendations that are popular among the group's members. It's useful on sites with a large but diverse readership to deliver rapid recommendations for a user who has limited information.

By Association: This is a subset of the previous category, commonly known as "Users who looked at X also looked at Y." It's as simple as looking at purchasing sequences or purchasing groups and providing related content to implement this type of recommendation system. This method is effective for obtaining recommendations for organically complementary content and at a certain moment in the user's life.

The collaborative filtering strategy has the advantage of not relying on machine-processable content, which allows it to accurately recommend complicated objects like movies without requiring a "knowledge" of the item. In recommender systems, many algorithms have been employed to measure user or item similarity. Consider the k-nearest neighbor (k-NN) method and the Pearson Correlation, both of which were pioneered by Allen. When creating a model from a user's behaviour, it's common to distinguish between explicit and implicit data collection methods.

Three issues plague collaborative filtering methods: cold start, scalability, and sparsity.

• Cold start: There isn't enough data to generate accurate recommendations for a new user or item.

• Scalability: There are millions of users and goods in many of the environments where these systems generate suggestions. As a result, calculating recommendations frequently necessitates a significant amount of computing resources.

• sparsity: The quantity of things available for purchase on major e-commerce sites is enormous. Only a small portion of the overall database will have been rated by the most active users. As a result, even the most popular things have a small number of ratings.

Item-to-item collaborative filtering (those who buy x also buy y), popularized by Amazon.com's recommender system, is one of the most well-known examples of collaborative filtering. By studying the network of connections between a user and their friends, several social networks used collaborative filtering to recommend new friends, groups, and other social connections. In hybrid systems, collaborative filtering is still used.

3.Hybrid Approach

For proposing products or items to the user, recommendation systems are widely utilised in a number of applications. Content-based and collaborative filtering are two prevalent ways for filtering recommendations. When there isn't enough data to learn the relationship between the user and the items, these solutions run into problems. In such instances, the Hybrid Recommendation System, a third type of technique, is utilised to develop recommendation systems. The disadvantages of both content-based and collaborative filtering methods are overcome with this approach.

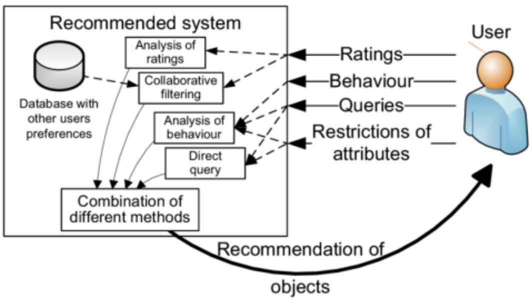

Figure 3 Hybrid Approach

A hybrid recommender system is one that generates an output by combining different recommendation algorithms. When comparing hybrid recommender systems to collaborative or content-based systems, hybrid systems typically have superior recommendation accuracy. The reason for this is a lack of knowledge about collaborative filtering's domain dependencies and people's preferences in a content-based system. When you combine the two, you get more common information, which helps you make better recommendations. Exploring novel ways to enhance core collaborative filtering algorithms with content data and content-based algorithms with user behaviour data is especially intriguing as knowledge grows. either incorporating content-based capabilities into a collaborative-based strategy (and vice versa); or by combining the techniques into a single model (for a comprehensive analysis of recommender systems, see []). Several studies comparing the efficacy of hybrid methods to pure collaborative and content-based methods have shown that hybrid methods can deliver more accurate suggestions than pure approaches. These strategies can also be utilised to solve difficulties like cold start and sparsity in recommender systems, as well as the knowledge engineering bottleneck in knowledge-based approaches. Netflix is an excellent illustration of how hybrid recommender systems may be used.

The website provides recommendations by comparing the viewing and searching habits of similar users (collaborative filtering), as well as by comparing the viewing and searching habits of different users. recommending films that have qualities in common with films that a user has rated highly (content-based filtering).

Related Work

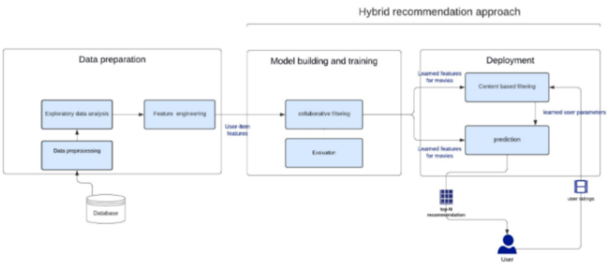

Figure 4 Demonstrating Hybrid Approach

```
In [11]: query = '''
         SELECT title, count(movieid) AS movies_count
         FROM MoviesWithRatings
         GROUP BY title,movieid
         ORDER BY movies_count DESC
         LIMIT 10;
         '''
         query_result=db.run_query(query)
         query_result
Out[11]:
```

	title	movies_count
0	Forrest Gump (1994)	329
1	Shawshank Redemption, The (1994)	317
2	Pulp Fiction (1994)	307
3	Silence of the Lambs, The (1991)	279
4	Matrix, The (1999)	278
5	Star Wars: Episode IV - A New Hope (1977)	251
6	Jurassic Park (1993)	238
7	Braveheart (1995)	237
8	Terminator 2: Judgment Day (1991)	224
9	Schindler's List (1993)	220

```
In [12]: plt.figure(figsize=(15, 8))
         barh = plt.barh(query_result['title'], query_result['movies_count'], color='blue')
         plt.title(' top ten most rated movies ')
         plt.xticks(np.arange(0, 400, 50))
         plt.tight_layout()
```

```
In [14]: for genre in genres_list:
             query = '''
             SELECT count(Movies.movieId)
             FROM  Movies
             where Movies.genres LIKE '%'''+genre+'''%'
             '''
             query_result=db.run_query(query)
             genres_count.append(query_result.iloc[0]['count'])

In [15]: data=pd.DataFrame({'x': genres_list, 'y': genres_count})

In [16]: data=pd.DataFrame({'x': genres_list, 'y': genres_count})
         plt.subplots(figsize=(10,5))
         sns.barplot(x=data.x, y=data.y, palette="rocket")
         plt.xticks(rotation=-45)
         plt.title('Most popular genres among directors');
```

Exploratory Data Analysis

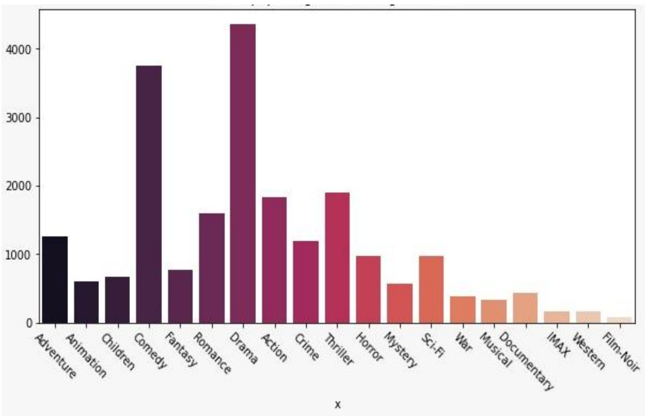

Figure 6 Most Popular Genres

```
In [27]: query = '''
SELECT rating, count(rating)
FROM Rating
GROUP BY rating
ORDER BY rating
'''
rating_result=db.run_query(query)
y = np.array(rating_result['count'])
mylabels = rating_result['rating']
explode =[0.3,0.1,0.1,0.1,0.1,0.1,0.2,0.1,0.1,0.1]
textprops = {"fontsize":15}
plt.pie(y, labels = mylabels,explode = explode,shadow = True,radius = 3,textprops =textprops)
plt.legend(title = "Rating",loc="lower left")
plt.title("Rating distribution for the Movies",fontsize=15)
plt.show()
```

Figure 7:Most popular genres

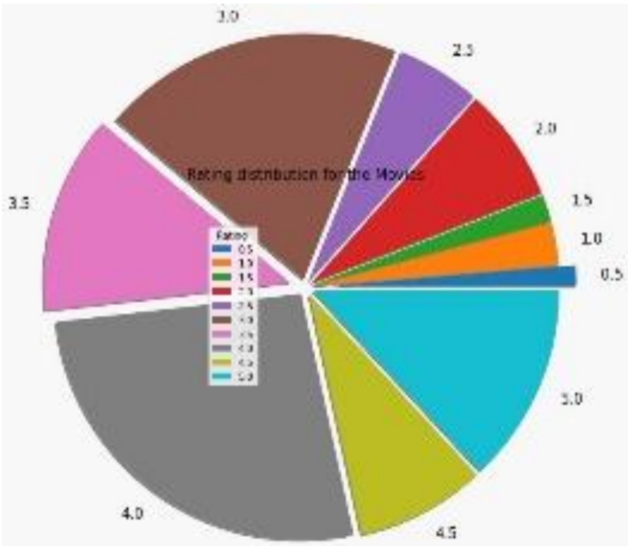

Figure 8 Rating Distribution for the movies

```
my_ratings = np.zeros((9724,1))

# test rating by web-applicaiton user
my_ratings[0] = 4
my_ratings[97] = 2
my_ratings[6] = 3
my_ratings[11]= 5
my_ratings[53] = 4
my_ratings[63]= 5
my_ratings[65]= 3
my_ratings[68] = 5
my_ratings[82]= 4
my_ratings[225] = 5
my_ratings[354]= 5

print("New user ratings:\n")
for i in range(len(my_ratings)):
    if my_ratings[i]>0:
        print("Rated",int(my_ratings[i]),"for Movie")
        print((moviesdataset.iloc[i]).title)
```

```
New user ratings:

Rated 4 for Movie
toy story
Rated 3 for Movie
sabrina
Rated 5 for Movie
dracula: dead and loving it
Rated 4 for Movie
indian in the cupboard, the
Rated 5 for Movie
fair game
Rated 3 for Movie
misérables, les
Rated 5 for Movie
screamers
Rated 4 for Movie
vampire in brooklyn
Rated 2 for Movie
braveheart
Rated 5 for Movie
little women
Rated 5 for Movie
above the rim
```

Figure 9 Testing Hybrid Recommendation

Conclusion

In this paper, we introduce a movie recommendation recommender system. Recommendation systems can be an effective approach to expose consumers to information they might not have discovered otherwise, which can help businesses achieve wider goals like increasing sales, ad income, or user engagement. However, there are a few crucial aspects to remember if you want to succeed with recommendation systems. Above all, recommendation systems should be necessary. It allows a user to choose from a variety of attributes and then recommends a movie list to him based on the cumulative weight of the various attributes and various algorithms. Due to the nature of our system, evaluating performance is difficult because there is no right or incorrect recommendation; it is simply a matter of opinion.Recommendation systems must be flexible as well. That is, versatile and able to change as the needs of the users change. Putting a recommendation system into production isn't the end of the process; rather, it's a continuous process of learning what works and what doesn't, as well as considering new data sources that could aid in making better recommendations.

References:

1. M Viswa Murali; T G Vishnu; Nancy Victor: "A Collaborative Filtering based Recommender System for Suggesting New Trends", 06 June 2019.
2. Bo Song, Yue Gao and Xiao-Mei Li: "Research on Collaborative Filtering Recommendation Algorithm Based on Mahout and User Model" 18 June 2018.
3. Ruiyang Wang and Tao Li – "Collaborative Filtering Recommender Systems", 12 August 2019.
4. Ashrita Kashyap, Sunita. B, Sneh Srivastava, Aishwarya, Anup Jung Shah: "A Movie Recommender System: MOVREC using Machine Learning Techniques", 6 May, 2018.

5. Collaborative Filtering - ML - GeeksforGeeks
6. Prem Melville and Vikas Sindhwani, Recommender Systems, Encyclopedia of Machine Learning, 2010.