Movie Recommendation System Using Machine Learning

Pooja Dhakane[1], Sayali Pokharkar[2], Kiran Thombare[3]

Electronic and Telecommunications Department,
MKSSS's Cummins College Of Engineering For Women Pune

pooja.dhakane@cumminscollege.in
sayali.pokharkar@cumminscollege.in
kiran.thombare@cumminscollege.in

Abstract- **Algorithms are used in recommendation systems to provide useful product and service suggestions to clients. Recently, those structures have been built using artificial intelligence, machine learning, and deep learning techniques. The recommendation system is divided into four categories: simple system based on popularity, collaborative filtering recommendation system, content-based completely recommendation system, and hybrid-based completely recommendation system. Because of its power in delivering more beneficial amusement, a film recommendation is becoming fairly diagnosed as part of our social existence. Such a machine will predict what films a person will enjoy based on a variety of factors such as the characteristics of previously enjoyed films by that person, the recognition of the movie, or keywords. Although a rigid structure for film advice has been proposed and implemented. Such a machine will predict what films a person will enjoy based on a variety of factors such as the characteristics of previously enjoyed films by that person, the recognition of the movie, or keywords. Although a rigid structure for film advice has been proposed and implemented.**

***Keywords*: Machine Learning, Movie Recommendation System,**
Content-Based, Collaborative filtering and Hybrid Approach.

I. INTRODUCTION

Every day, technological progress reaches new heights, and as a result, statistics will rise dramatically. The superfluous statistics from a recommender system are removed using information filtering structures. Supervised Learning, Unsupervised Learning, and Reinforcement Learning are the three broad areas through which you can learn about gadgets. Collaborative filtering and content-based completely filtering are both used in recommender systems. A content-based completely filtering strategy makes use of a series of specific, pre-tagged qualities of an object to recommend further objects with similar properties. The role of a film recommender is unique Because of its ability to provide better entertainment, it has become a part of our social lives.

II. A HISTORY OF RECOMMENDER SYSTEMS

Recommender systems have become increasingly common in recent years, and they are used in a wide range of applications. Films, music, news, publications, research articles, search queries, social tags, and products in general are the most well-known.

In 1990, it appeared that the first attempt was made to build an advising machine. MAFIA became an agent for email filtering as part of a larger intelligent report processing assistance machine. It was revolutionary at the time because it was able to get around the limitations of the existing message filtering structures: the demand for pre-defined text. It became capable of recognising and extracting relevant ideas from unstructured information, as well as constructing a semantic depiction that may assist the user to quickly locate the information required

Movie Lens, a website that recommends unique films to its users based entirely on their tastes, was one of the first attempts in film recommendation. Amazon is probably the most well-known recommender system these days. Item-to-Item Collaborative Filtering is used by the Amazon Recommender Machine.

One occurrence in recent years that aided the growth of interest in recommender systems was the Netflix Prize, a Netflix-sponsored competition that took place from 2006 to 2009.

III. DEFINITION

A recommendation system's principal goal is to make meaningful object recommendations to a set of clients. A Recommendation System is a subtype-like data structure filtering system that attempts to anticipate the ratings a customer would give an item. Recommender systems are used in many areas and are most commonly identified as playlist creators for video and music services such as Netflix, YouTube, and Spotify, Recommendation Systems for services such as Amazon, or content-based recommenders for social media platforms such as Facebook, Instagram, and Twitter, among others.

IV. METHODS

In the field of machine learning, Recommendation Systems are classified as follows:

1. Collaborative Filtering Recommendation System
2. Content-based Recommendation System
3. Hybrid-based Recommendation System
4. Knowledge-based Recommendation System
5. Utility-based Recommendation System
6. Demographic-based Recommendation System

1. Collaborative Filtering Recommendation System

It discovers comparable users and recommends what they enjoy in the Collaborative Filtering Recommendation System. This form of recommendation system does not utilise the item's features to promote it; instead, it groups users into similar types of Clusters and then recommends items to each user based on their cluster's preferences.

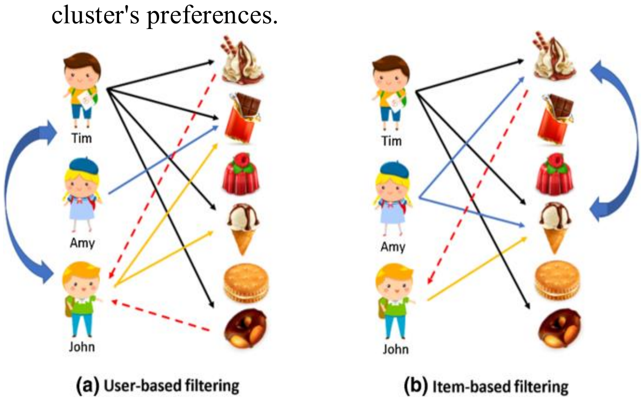


Fig 1- Collaborative Filtering Recommendation

1) **Scalability:** Because the number of users and objects in the recommendation system is growing all the time, data processing could take a long time. As a result, the time it takes for users to respond may lengthen. As a result, the only solution is to process the data offline at regular intervals, leaving only voting online. Voting takes less time.
2) **Sparsity:** Despite the fact that the system works with a considerable volume of data, the number of active users is small. As a result, finding a common set of goods for similar users is tough. In other words, two comparable users can't help each other predict because they haven't rated the same things. As a result, the recommender system's efficiency suffers.
3) **Cold-Start Problem:** Because the number of users and products in the recommendation system is growing all the time, processing the data could take a long time. As a result, the time it takes for users to respond may rise. As a result, the only solution is to process the data offline at regular intervals, leaving only voting to be done online. Voting takes less time than other processes.

2. Content-based Recommendation

A Content-Based Recommender generates a personal profile based on the data we gather from the individual, either explicitly or implicitly, and then uses that profile to indicate to the person that they have given or taken extra effort. If the user follows the advice, the engine will get more precise.

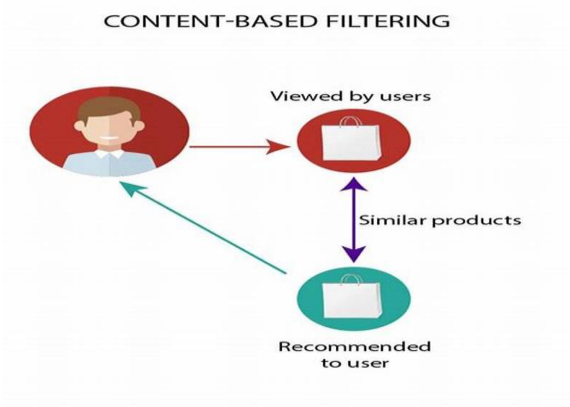


Fig 2- Content-based Recommendation

COSINE SIMILARITY:

The angle of cosine between two items is measured by cosine similarity between two things. It compares files on a scale that is normalised. It can be accomplished by locating the dot product between the two identities.

3. Hybrid-based Recommendation

A hybrid recommendation system is the only one that combines many different guidance methodologies to provide one result. When comparing hybrid recommender structures to collaborative or content-based structures, the advice accuracy in hybrid structures is usually higher.

The reason for this is a lack of information on collaborative filtering's spatial dependencies and people's options in content-based completely machines. The sum of each produces an unusually large amount of information, which helps to boost recommendations.

4. Knowledge-based Recommendation

When a recommender system offers recommendations based on particular queries rather than a user's rating history, it is said to be knowledge-based. It may ask the user to provide a set of criteria or guidelines for how the results should appear, as well as an example of an item. The system then searches its item database for similar things and returns them.

5. Utility-based Recommendation

A utility-based recommender system produces recommendations for the user based on the utility of each object. Creating a utility for individual users is, of course, the central difficulty for this type of system. The key benefit of utilising a utility-based recommender system is that it may consider non-product variables into the utility calculation, such as vendor reliability and product availability. This allows the user to view the object's current inventory in real time.

V. *RELATED WORK*

For this project, we will consider the two Dataset. One is tmdb_5000_movies.csv and tmdb_5000_credits.csv. These Dataset is taken from Kaggle.

The tmdb_5000_movies.csv consists of following columns:

1. Budget
2. Genres
3. Homepage
4. Id

5. Keywords
6. original_language
7. original_title
8. overview
9. popularity
10. production_companies
11. production_countries
12. release_date
13. revenue
14. runtime
15. spoken_languages
16. status
17. tagline
18. title
19. vote_average
20. vote_count

The tmdb_5000_credits.csv consists of following columns:

1. movie_id
2. title
3. cast
4. crew

These two Dataset we combine together on the basis of 'title' column. But for our project we consider following columns mainly:

1. movie_id
2. title
3. overview
4. genres
5. keywords
6. cast
7. crew

VI. RESULT AND DISCUSSION

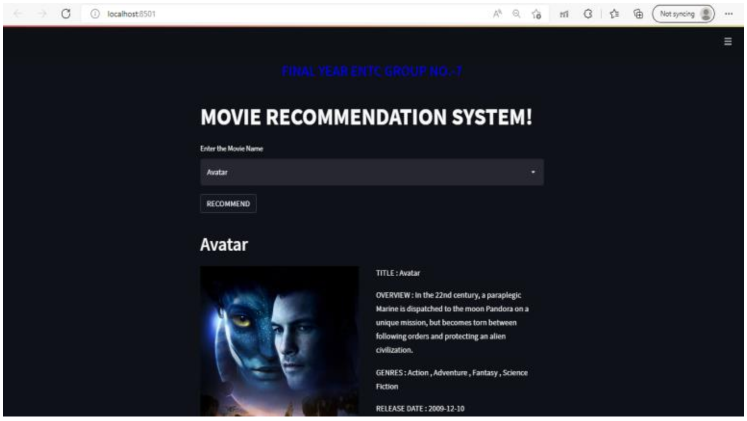
In the machine learning framework recommendation system, we have used Cosine Similarity and Content-based filtering to predict our result and recommend a movie to the user by running the code in Jupyter Notebook and Pycharm.

Experiment Result:

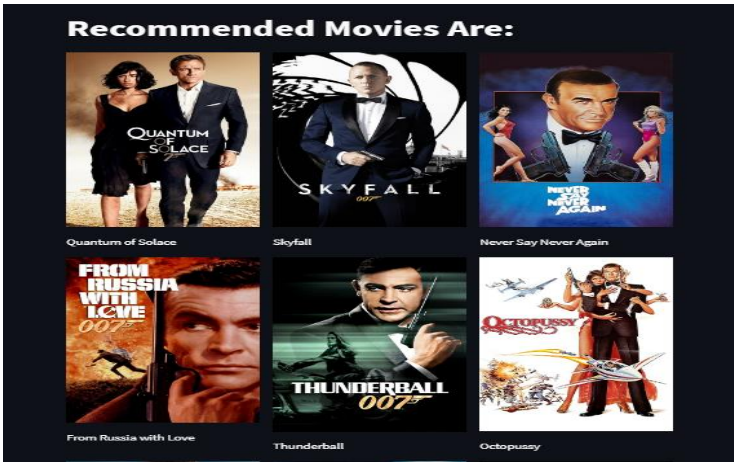
A. Result on Jupyter Notebook

```
In [49]: recommendation_system('Spectre')

Top 9 recommended movies based on your choice:

1 -> Quantum of Solace
2 -> Skyfall
3 -> Never Say Never Again
4 -> From Russia with Love
5 -> Thunderball
6 -> Octopussy
7 -> Safe Haven
8 -> Diamonds Are Forever
9 -> Licence to Kill
```

B. Result on Local Server



C. Result On Local Server with Recommendation



D. URL to run Project on Remote Server

https://movie-recommender-systemgroup7.herokuapp.com/

Experiment Discussion:

The formula used to measure how similar the movies are based on their similarities of different properties. Mathematically, it shows the cosine of the angle of two vectors projected in a multidimensional space. The cosine similarity is very beneficial since it helps in finding similar objects.

$$cosine(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Fig 3- Cosine Similarity Formula

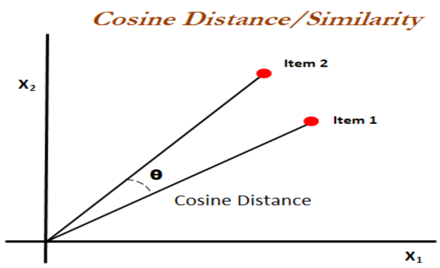


Fig 4- Cosine similarity

The angle theta between the two movies will determine the similarity between the two movies. The theta ranges from 0- 1. If the value of the theta is near 1 then it is most similar and if it's near to 0 then it is least similar. The movie will be recommended if it is close to 1 otherwise there would be no similarity between them. It will recommend the best movies to the user according to the Cosine similarity. After the cosine similarity, we have used a normalised popular score through which we get our function of computing distance. Then by using the KNN functionality, we have found the nearest neighbour which will be recommended to the user.

VII. CONCUSION

We have concluded from the movie recommender system by making the use of content-based filtering in the movie recommendation system. The KNN algorithm is implemented in this model along with the principle of cosine similarity as it gives more accurate results than the other distance metrics and the complexity is comparatively low as compared to another algorithm. Recommender systems have become the most essential fount of a relevant and better source of information in the Internet's world. Simple ones consider one or a few things while the more complex ones make use of more things to filter the results and make them more user-friendly.

With the inclusion of advanced deep learning and other filtering techniques like collaborative filtering and hybrid filtering a strong movie recommendation system can be built. This can be a major step towards the further development of this model as it will not only become more efficient to use but also increase the business value even further.

VIII. REFERENCES

1. Charu C. Aggarwal's A Recommender System Textbook

2. "Movie recommendation system employing cosine similarity and knn," R. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav.

3. "Article: A movie recommender system: Movrec," International Journal of Computer Applications, M. Kumar, D. Yadav, A. Singh, and V. K. Gupta.

4. N. Hug, Journal of Open Source Software, "Surprise: A Python Library for Recommender Systems."

5. L. M. de Campos, J. M. Fernandez-Luna, J. F. Huete, and M. A. Rueda-Morales, "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks," International Journal of Approximate Reasoning.

6.https://www.slideshare.net/CrossingMinds/recommendation-system-explained

7.https://www.mygreatlearning.com/blog/masterclass-on-movie-recommendation-system/