Evaluation Report on Recommender Systems Applied.

Introduction:-

A recommendation system is an information filtering system that provides users personalized recommendations based on their preferences and behavior. These systems are commonly used in e-commerce platforms, social media, music and video streaming, and other applications where a large amount of content is available. It can be difficult for users to find what they are looking for. Recommender systems analyze data about user behavior, such as search queries, browsing history, and purchase history, as well as data about available products, such as product descriptions, user ratings, and ratings. Overall, recommendation systems are invaluable tools for businesses and consumers, enabling more personalized and efficient interactions with different content and products.

Building Recommendation Systems:-

- Recommendation systems provide personalized recommendations that benefit businesses and consumers.
- Creating a recommendation system involves collecting and pre-processing data on user behavior and potential products.
- The next step is to choose the correct recommendation algorithm, which depends on the specific system requirements and data characteristics.
- Model training and optimization involve splitting the data into training and validation sets and optimizing model performance.
- Deploying the model to a production environment includes integrating it into the user interface and back-end systems, monitoring its performance, and making necessary changes.
- Privacy and ethical implications must be considered when developing a recommendation system, such as transparency, explainability, and respecting user privacy and autonomy.

Evaluating the Recommendation Systems

Collaborative Recommendation System for Movies

Collaborative filtering is a recommender system that predicts a user's interests or preferences based on the behavior and preferences of similar users. Collaborative filtering identifies users who interact with the system and recommends items that similar users have preferred.

Collaborative filtering can be based on explicit or implicit feedback. Explicit feedback is when users rate or review products, while implicit feedback is when the system infers user preferences based on their behavior, such as browsing history or purchase history.

There are two main types of collaborative filtering: by the user and by items. User-based collaborative filtering recommends items to a user based on the preferences of other similar users. In contrast, item-based collaborative filtering recommends items to the user based on their similarity to items they have interacted with in the past.

Overall, Collaborative filtering is a popular and effective method for recommender systems that can generate accurate and personalized user recommendations based on similar users behavior and preferences.

➢ Dataset Used

The Full MovieLens Dataset contains metadata for 45,000 movies released on or before July 2017, including information on cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDB vote counts, and vote averages. The dataset also includes files containing 26 million ratings from 270,000 users on all 45,000 movies, with ratings on a scale of 1-5 obtained from the official Group Lens website. The dataset is comprised of several files, including movies_metadata.csv, keywords.csv, credits.csv, links.csv, links_small.csv, and ratings_small.csv. The Full MovieLens Dataset can be accessed separately and includes 26 million ratings and 750,000 tag applications from 270,000 users on all 45,000 movies in the dataset.

Dataset Link:-
https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?select=movies_metadata.csv

Model Based Recommender Systems

Model-based recommender systems use machine learning models to generate personalized recommendations for users based on historical user-item interaction data. These models use matrix factorization techniques to identify latent features that explain the observed user-item interactions. The models can then use these latent features to make personalized recommendations for users. Model-based recommender systems involve building a model based on the dataset of ratings, allowing for more efficient and scalable recommendations.

For model-based recommender systems, we are going to use a library called Surprise, and we are going to use SVD as a matrix factorization method.

Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is one of the matrix factorization methods in machine learning. Singular value decomposition decomposes a matrix into three other matrices and extracts the features from the factorization of a high-level (user-item-rating) matrix.

The formula of SVD can be given as -

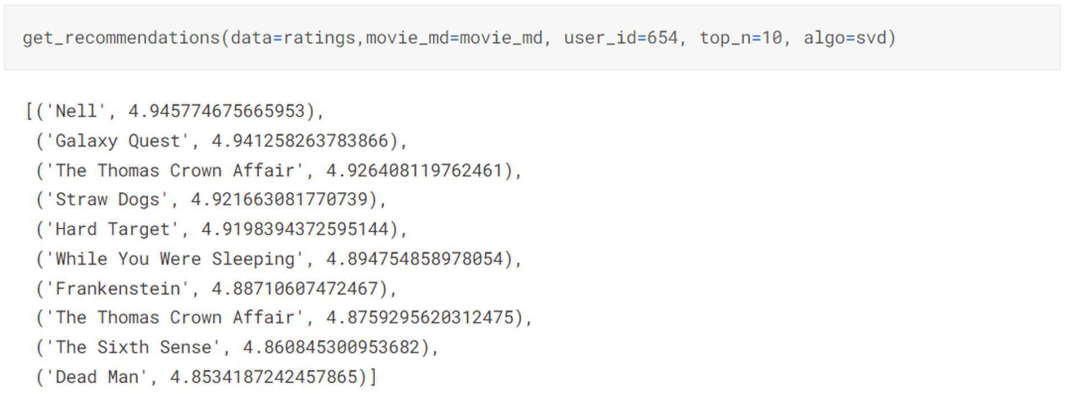
$$A = USV^T$$

Where,

Matrix U: Latent features of Users

Matrix S: Diagonal matrix representing the strength of each feature

Matrix U: Latent features of Items



```
get_recommendations(data=ratings,movie_md=movie_md, user_id=654, top_n=10, algo=svd)
```

```
[('Nell', 4.945774675665953),
 ('Galaxy Quest', 4.941258263783866),
 ('The Thomas Crown Affair', 4.926408119762461),
 ('Straw Dogs', 4.921663081770739),
 ('Hard Target', 4.9198394372595144),
 ('While You Were Sleeping', 4.894754858978054),
 ('Frankenstein', 4.88710607472467),
 ('The Thomas Crown Affair', 4.8759295620312475),
 ('The Sixth Sense', 4.860845300953682),
 ('Dead Man', 4.8534187242457865)]
```

Output for Model Based Recommender Systems

Model-based recommender systems have several advantages and disadvantages withrespect to collaborative filtering Here are some pros and cons:

Advantages:

1. Personalized recommendations: Model-based systems can create more personalized recommendations for users by capturing complex data patterns of user-item interactions and using many latent attributes to determine the underlying structure of the data
2. Scalability: Model-based systems are more scalable than memory-based systems because they can use matrix factoring techniques to efficiently identify latent characteristics of the data, rather than calculating the similarities of all pairs of usersor items
3. Higher Accuracy: Model-based systems can often achieve higher accuracy than memory-based systems because they can capture more complex relationships between users and elements.
4. Ability to handle sparseness: Model-based systems can handle sparse data more efficiently than memory-based systems because they can identify latent characteristics of the data even if there are missing values in the user element interaction matrix.

Disadvantages:

1. Cold start problem: Model-based systems are often less effective when there is littledata on new users or articles, as a lot of historical interaction data is needed to generate accurate recommendations

2. Interpretability: Model-based systems are often less interpretable than memory- based systems because they rely on latent factors that are difficult to understand orinterpret
3. Algorithm complexity: Model-based systems can be more complex to implement andrequire more processing resources than memory-based systems because they use machine learning models to identify latent data properties.
4. Need for parameter tuning: Model-based systems require hyperparameter tuning, such as the number of latent functions, for optimal performance, which can be time-consuming.

Memory Based Recommender System

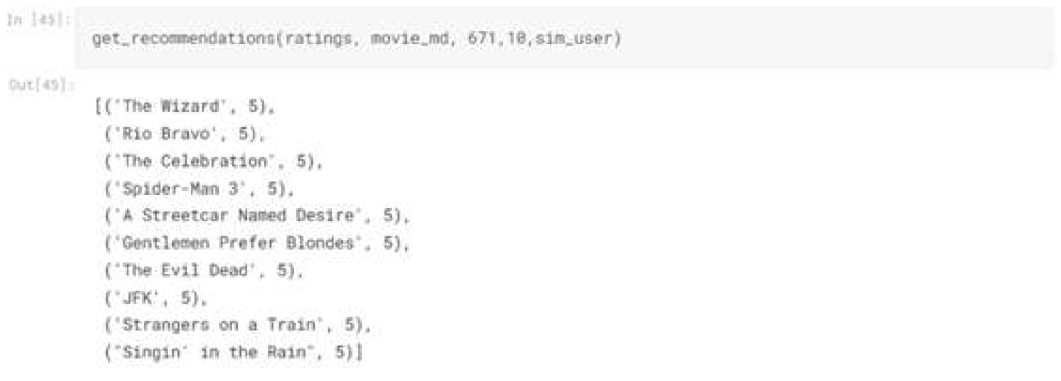
Memory-based recommendation systems are a type of collaborative filtering system that use the user-item interaction data to generate recommendations for users. Unlike model- based approaches that use machine learning models to identify latent features that explain the user-item interactions, memory-based approaches rely on similarities between users or items to generate recommendations.

There are two main types of memory-based recommendation systems: user-based anditem-based.

User-based recommendation system:

In a user-based recommendation system, the system identifies a group of users who have similar preferences to the user and generates recommendations based on the items they have liked or interacted with. The system computes the similarity between users based on their interaction patterns and then recommends items that similar users have liked.

For example, if a user has liked several movies from the action genre, the system might identify other users who have also liked those movies and recommend other action moviesbased on their preferences.



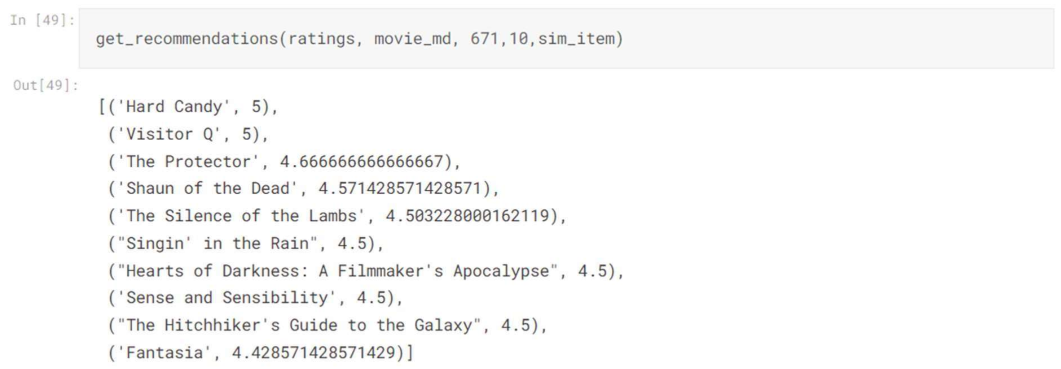
```
In [45]:  get_recommendations(ratings, movie_md, 671,10,sim_user)

Out[45]:  [('The Wizard', 5),
           ('Rio Bravo', 5),
           ('The Celebration', 5),
           ('Spider-Man 3', 5),
           ('A Streetcar Named Desire', 5),
           ('Gentlemen Prefer Blondes', 5),
           ('The Evil Dead', 5),
           ('JFK', 5),
           ('Strangers on a Train', 5),
           ("Singin' in the Rain", 5)]
```

Output for User-based recommendation system

Item-based recommendation system:

In an item-based recommendation system, the system identifies a group of items that are similar to the items that the user has liked or interacted with in the past, and generates recommendations based on those items. The system computes the similarity between itemsbased on the patterns of user interactions with those items and then recommends similar items to the user.

For example, if a user has liked several romantic comedies, the system might identify otherromantic comedies that are similar to the ones the user has liked, and recommend those movies to the user.



```
In [49]:  get_recommendations(ratings, movie_md, 671,10,sim_item)

Out[49]:
          [('Hard Candy', 5),
           ('Visitor Q', 5),
           ('The Protector', 4.666666666666667),
           ('Shaun of the Dead', 4.571428571428571),
           ('The Silence of the Lambs', 4.503228000162119),
           ("Singin' in the Rain", 4.5),
           ("Hearts of Darkness: A Filmmaker's Apocalypse", 4.5),
           ('Sense and Sensibility', 4.5),
           ("The Hitchhiker's Guide to the Galaxy", 4.5),
           ('Fantasia', 4.428571428571429)]
```

Output for Item-based recommendation system

Memory-based recommender systems have several advantages and disadvantages Here aresome:

Advantages:

1. Easy to implement: Memory-based systems are relatively easy to implement because they do not require complex algorithms or large computing resources
2. Real-time recommendations: Memory-based systems can generate real-time recommendations because they do not require pre-processing and machine learningmodel training.
3. Transparency: Memory-based systems are often more transparent than model- based systems because they generate recommendations based on user or item similarities that can be easily understood and interpreted
4. Interpretability: Memory-based systems are often more interpretable than model- based systems because they rely on user-element interaction data that can be easilyviewed and understood Before.

Disadvantage:

1. Limited scalability: Memory-based systems can become expensive as the size of user-item interaction data increases, as they require computational similarities between all pairs of users or items.
2. Limited coverage: Memory-based systems may not be able to generate recommendations for all users or items in a dataset because they rely on user-iteminteraction data.
3. Limited personalization: Memory-based systems may generate less personalized recommendations than model-based systems because they are based on user or item similarities and may not capture complex user-item interaction data patterns.
4. Cold start problem: Memory-based systems can be less effective when there is littledata about new users or articles because they need historical interaction data to generate accurate recommendations.

Content-Based Based Recommendation System

Content-based filtering is a type of recommendation system that suggests items to users based on the characteristics or features of the items themselves. This approach involves analyzing attributes such as genre, ratings, or attribute contents of items that users have interacted with in the past and generating recommendations for similar items. Content-based filtering is particularly useful for recommending less popular or niche items, and it can be effective for users with unique preferences. However, it may struggle to recommend new or diverse items that do not share many features with past interactions, and it may not capture more complex aspects of user preferences. Overall, content-based filtering is an effective approach to recommendation systems that generates personalized recommendations based on the features of past interactions.

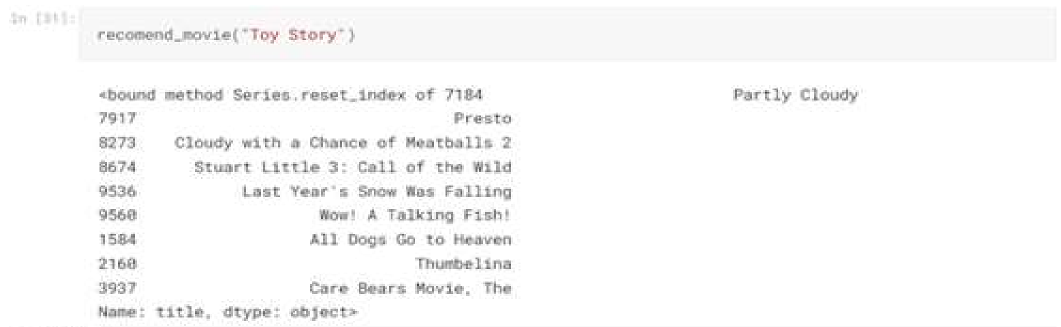
Dataset Used:-

The ml-latest-small dataset provides information on 100836 movie ratings and 3683 tag applications from MovieLens, a movie recommendation service. The data spans 9742 movies and were created by 610 users between March 29, 1996 and September 24, 2018. All selected users had rated at least 20 movies, but no demographic information is provided. The dataset includes four files: links.csv, movies.csv, ratings.csv, and tags.csv.

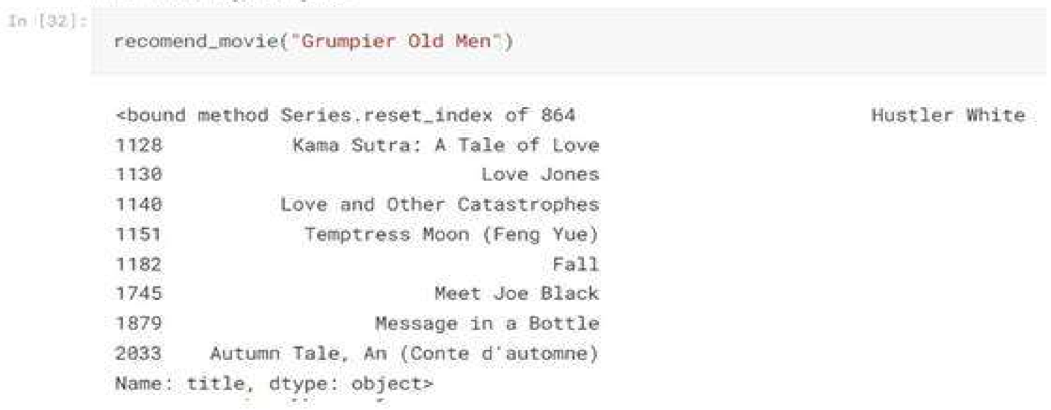
Dataset Link:-

https://www.kaggle.com/datasets/shubhammehta21/movie-lens-small-latest-dataset

Content-Based Based Recommendation System Output:-



```
In [31]:
recomend_movie("Toy Story")

<bound method Series.reset_index of 7184                      Partly Cloudy
7917                         Presto
8273    Cloudy with a Chance of Meatballs 2
8674      Stuart Little 3: Call of the Wild
9536           Last Year's Snow Was Falling
9560                 Wow! A Talking Fish!
1584                All Dogs Go to Heaven
2160                         Thumbelina
3937                Care Bears Movie, The
Name: title, dtype: object>
```



```
In [32]:
recomend_movie("Grumpier Old Men")

<bound method Series.reset_index of 864                      Hustler White
1128           Kama Sutra: A Tale of Love
1130                           Love Jones
1140          Love and Other Catastrophes
1151            Temptress Moon (Feng Yue)
1182                                 Fall
1745                       Meet Joe Black
1879                  Message in a Bottle
2033     Autumn Tale, An (Conte d'automne)
Name: title, dtype: object>
```

Output for Content Based Recommendation System

Content-based filtering systems have several advantages and disadvantages. Here are someof the pros and cons:

Advantages:-

1. Personalization: Content-based systems can provide highly personalized recommendations, as they use a user's past interactions with items to generaterecommendations that match their preferences.
2. No cold start problem: Content-based systems do not suffer from the cold startproblem, as they do not require historical interaction data to generate recommendations.
3. Explanation: Content-based systems can provide an explanation for their recommendations, as they are based on the characteristics of items and can beeasily understood by users.
4. Diversity: Content-based systems can provide diverse recommendations, as they are able to identify items with similar characteristics that a user may not have discovered on their own.

Disadvantages:-

1. Limited recommendation scope: Content-based systems can only recommend itemsthat are similar to items a user has already interacted

with, which may limit the scope of recommendations.

2. Limited discovery of new items: Content-based systems may not be able to discover new items that a user has not previously interacted with, as they rely on a user's pastinteractions with items.
3. Limited coverage of user preferences: Content-based systems may not be able to capture all aspects of a user's preferences, as they rely on the content features of items and may not capture other factors that influence a user's preferences.
4. Over-specialization: Content-based systems may generate recommendations thatare too similar to the items a user has already interacted with, leading to over- specialization and a lack of diversity in recommendations.

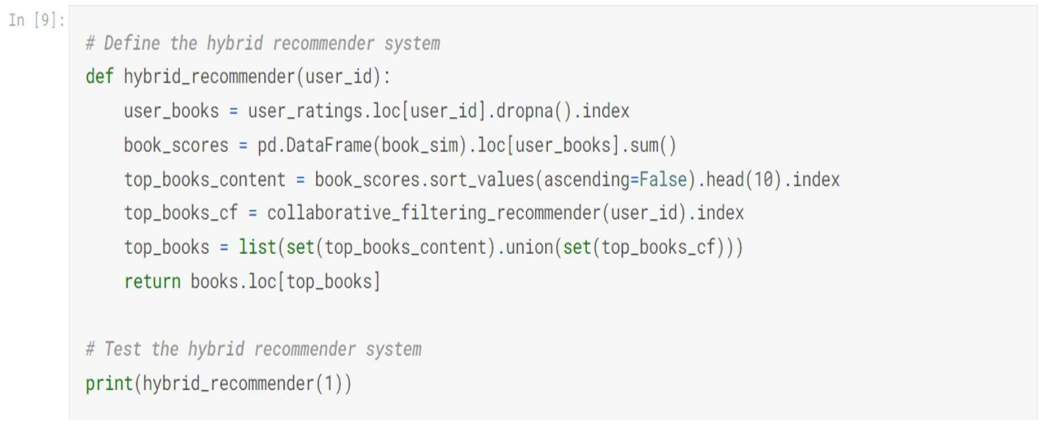
Hybrid Recommendation System

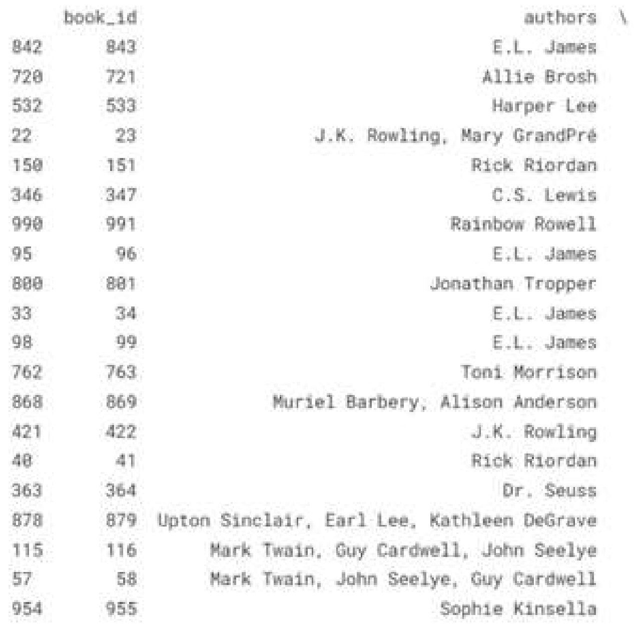
Hybrid recommender systems combine multiple recommendation approaches to provide more accurate and diverse recommendations for users. There are two types of hybrid systems: model-based and feature-based. Model-based systems combine different recommendation models, while feature-based systems use multiple item features. Hybrid systems can improve accuracy and provide more serendipitous recommendations. However, building a hybrid system can be challenging and computationally expensive.

Overall, hybrid recommender systems are a popular and effective approach to recommendation systems that can generate more accurate and diverse recommendations for users by combining multiple types of recommendation techniques.

Dataset Used:-

The data used is a books rating set which has csv like Average rating, Final data rating, Rating count, and Ratings which has following headers Book id, Rating, Author, Title, Genre,Rating count of individual.


```
In [9]:
# Define the hybrid recommender system
def hybrid_recommender(user_id):
    user_books = user_ratings.loc[user_id].dropna().index
    book_scores = pd.DataFrame(book_sim).loc[user_books].sum()
    top_books_content = book_scores.sort_values(ascending=False).head(10).index
    top_books_cf = collaborative_filtering_recommender(user_id).index
    top_books = list(set(top_books_content).union(set(top_books_cf)))
    return books.loc[top_books]

# Test the hybrid recommender system
print(hybrid_recommender(1))
```



```
     book_id                                        authors  \
842      843                                     E.L. James
720      721                                    Allie Brosh
532      533                                     Harper Lee
22        23                      J.K. Rowling, Mary GrandPré
150      151                                   Rick Riordan
346      347                                     C.S. Lewis
990      991                                 Rainbow Rowell
95        96                                     E.L. James
800      801                                Jonathan Tropper
33        34                                     E.L. James
98        99                                     E.L. James
762      763                                   Toni Morrison
868      869                  Muriel Barbery, Alison Anderson
421      422                                   J.K. Rowling
40        41                                   Rick Riordan
363      364                                      Dr. Seuss
878      879  Upton Sinclair, Earl Lee, Kathleen DeGrave
115      116        Mark Twain, Guy Cardwell, John Seelye
57        58        Mark Twain, John Seelye, Guy Cardwell
954      955                                Sophie Kinsella
```

Output for Hybrid Recommendation System

The Hybrid Recommendation system was developed with two main types of hybrid recommender systems: model-based and feature-based. Model-based hybrid systems combine multiple recommendation models, such as collaborative filtering and content- based filtering, into a single model that generates recommendations. Feature-based hybridsystems, on the other hand, use multiple features of items, such as genre, director, and actor, to generate recommendations.

Here are some of the pros and cons of hybrid recommender systems:

Advantages:

1. Improved accuracy: Hybrid systems can improve the accuracy of recommendationsby combining the strengths of multiple recommendation techniques.
2. Increased coverage: Hybrid systems can increase the coverage of recommendationsby using different techniques to recommend items that may not be captured by a single technique.
3. Improved diversity: Hybrid systems can improve the diversity of recommendations by combining multiple techniques that recommend items based on different aspects.
4. Effective in handling data sparsity: Hybrid systems can handle data sparsity better than individual recommendation techniques by leveraging the complementary information from different techniques.

Disadvantages:

1. Increased complexity: Hybrid systems are often more complex to develop and maintain than individual recommendation techniques, as they require integration of different recommendation techniques.
2. Increased computational requirements: Hybrid systems can require more computational resources than individual recommendation techniques, as they involve combining the results from multiple recommendation techniques.
3. Need for parameter tuning: Hybrid systems require tuning of hyperparameters to achieve optimal performance, which can be a time-consuming process.
4. Transparency: Hybrid systems may be less transparent than individual recommendation techniques, as they involve combining the results from multiple techniques and may not provide clear reasons for the recommendations.

Conclusion:-

All types of Recommendation Systems worked on this project are compared with their pros and cons. Based on requirements, we can use any of these approaches. In Content-Based Recommender, we built a system that took movie overview, taglines, metadata, cast, crew, genre, and keywords to make predictions using the cosine similarity of the data given. In Collaborative Filtering, we used the powerful Surprise Library to build a collaborative filter based on singular value decomposition. At last, we used a hybrid approach, a mix of context-based and collaborative Filtering, to implement the system. This approach overcomes each algorithm's drawbacks and improves the system's performance. Techniques like Similarity and Classification are used to get better recommendations, thus increasing precision and accuracy.