

CAPSTONE PROJECT

**Movies Recommender System**

**GROUP-2**

## -Final Report

**Mentored by: Submitted by:**

**Mr. Chandran Venkateshan Mohammad Kareem Khan Rishab Jain**

**Chayan Dwivedi**

**Deepshree Kondra**

**Kadakuzhill Job Himalaya**

**Sai Kumar**

**Table of Contents:**

|  |  |
| --- | --- |
| 1. Industry Review |  |
| 1.1 | Current Practices |
| 1.2 | Background Research |
| 1.3 | Literature Survey |
| 2. Dataset and Domain | |
| 2.1 | Data Attribute Details |
| 2.2 | DataPreprocessing Analysis |
| 2.3 | Project Justification |
| 3. Data Exploration |  |
| 3.1 | Univariate Analysis |
| 3.1.1 | Count of UserID Column |
| 3.1.2 | Count of MovieID Column |
| 3.1.3 | Count of MovieTitle Column |
| 3.1.4 | Count of Rating Column |
| 3.2 | Bivariate Analysis |
| 3.2.1 | Graph between different Genres and |
|  | Ratings |
| 1. Feature Engineering 2. Base Model    1. Future Work |  |

# Industry Review :-

### Current Practices:

-- Movie recommender systems are widely used today in various applications such as online movie streaming platforms, movie review websites, and social media platforms.

-- Personalized Recommendations: Movie recommender systems provide personalized recommendations to users based on their viewing history, preferences, and ratings.

-- Improved User Experience: Movie recommender systems enhance the user experience by helping users discover movies that align with their tastes and interests.

-- Increased Engagement: Recommender systems can increase user engagement by suggesting movies that are more likely to be of interest to them.

-- Customer Retention: Recommender systems help retain customers by offering Personalize recommendations that keep users coming back to the platform.

-- Cross-selling: Recommender systems can also be used for cross-selling by suggesting related movies or TV shows to users based on their viewing history.

### Background Research:

* Recommender systems are algorithms and techniques that are designed to help users find relevant items, products, or information in a large and complex dataset. They are used in a variety of domains, including e-commerce, social media, music and video streaming, and online advertising.
* There are several different types of recommender systems, including:

Collaborative filtering: This approach relies on the past behavior of users and item ratings to predict future preferences. Collaborative filtering is often used for recommending movies, music, and books.

Content-based filtering: This approach recommends items based on their attributes or features. For example, a music recommendation system might recommend songs with similar genres or lyrics.

Hybrid approaches: These combine multiple recommendation techniques to provide more accurate and diverse recommendations.

* There are several challenges associated with building recommender systems, including data sparsity, cold-start problems, and the "filter bubble" effect. Data sparsity occurs when there are many users and items but few interactions between them, which can make it difficult to generate accurate recommendations. Cold-start problems arise when there is not enough data about new users or items, making it challenging to make recommendations. The filter bubble effect refers to the tendency for recommendation systems to reinforce users' existing preferences and limit exposure to new or diverse content.

Recent advances in machine learning, deep learning, and natural language processing have led to the development of more sophisticated recommender systems that can better handle these challenges. These systems can take into account a wider range of user and item features, as well as user context and feedback, to provide more personalized and diverse recommendations.

4

### Literature Survey:

* + 1. **Publications :**

In the last 16 years, more than 200 research articles were published about research-paper recommender systems.We reviewed these articles and present some descriptive statistics in this paper, as well as a discussion about the major advancements and shortcomings and an overview of the most common recommendation concepts and approaches. We found that more than half of the recommendation approaches applied content-based filtering (55%).

Collaborative filtering was applied by only 18% of the reviewed approaches, and graph- based recommendations by 16%. Other recommendation concepts included stereotyping, item-centric recommendations, and hybrid recommendations. Our review revealed some shortcomings of the current research. First, it remains unclear which recommendation concepts and approaches are the most promising. For instance, researchers reported different results on the performance of content based and collaborative filtering.

Sometimes content-based filtering performed better than collaborative filtering and sometimes it performed worse. We identified three potential reasons for the ambiguity of the results. (A) Several evaluations had limitations. They were based on strongly pruned datasets, few participants in user studies, or did not use appropriate baselines. (B) Some authors provided little information about their algorithms, which makes it difficult to re- implement the approaches. Consequently, researchers use different implementations of the same recommendations approaches, which might lead to variations in the results. (C) We speculated that minor variations in datasets, algorithms, or user populations inevitably lead to strong variations in the performance of the approaches. Hence, finding the most promising approaches is a challenge.

### Application :

Recommender systems have a wide range of applications in various industries, including:

**E-commerce**: Recommender systems are commonly used in online marketplaces like Amazon, eBay, and Walmart to suggest products to users based on their past browsing and purchase history.

**Entertainment**: Streaming platforms like Netflix, Hulu, and Spotify use recommender systems to suggest movies, TV shows, and music to users based on their viewing and listening history.

**Social media**: Social media platforms like Facebook and Instagram use recommender systems to suggest content and pages to users based on their interests and engagement history.

**Travel**: Travel websites and booking platforms like Expedia, Booking.com, and TripAdvisor use recommender systems to suggest hotels, flights, and activities to users based on their past bookings and preferences.

**Healthcare**: Recommender systems are used in healthcare to suggest treatments and medications based on a patient's medical history and symptoms.

**Education**: Recommender systems are used in online learning platforms to suggest courses and learning materials to students based on their previous learning history and preferences.

**News**: News websites use recommender systems to suggest articles to readers based on their reading history and interests.

### Undergoing research :

Recommender systems have been an active research area for many years, and there are many avenues for exploration depending on your interests and goals. Here are some potential research topics related to recommender systems:

New recommendation algorithms: There are many different types of recommendation algorithms, including collaborative filtering, content-based filtering, and hybrid approaches. Developing new recommendation algorithms or improving upon existing ones is an active area of research.

Personalization: Personalization is a key feature of recommender systems. Exploring new ways to personalize recommendations for users, such as using social network data, location- based data, or demographic data, is a promising area of research.

# Dataset and Domain :-

### Data Attribute Details:

In the dataset, we will encode all the categorical values into Numerical Values as shown.

|  |  |
| --- | --- |
| UserID | ID of the User |
| MovieID | ID of the Movie |
| Rating | Movie rating given by the user |
| Timestamp | Duration of the movie |
| Title | Movie Name |
| Genres | Different genres of movies |

### Pre-processing Data Analysis:

Range Index: 100836 entries (total 5 columns):

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No | Variables Names | Categorization of Variable | Null values Check |
| 1. | UserID | Categorical | 100836 non\_null object |
| 2. | MovieID | Categorical | 100836 non\_null object |
| 3. | Rating | Categorical | 100836 non\_null object |
| 4. | Timestamp | Numerical /Discrete | 100836 non\_null object |
| 5. | Title | Categorical | 100836 non\_null object |
| 6. | Genres | Categorical | 100836 non\_null object |

In this dataset , we don’t have any null values in the dataset hence the dataset is free from null values.

### Project Justification :-

#### Problem Statement:-

Develop a movie recommender system that can suggest movies to users based on their preferences, historical viewing habits. The system should provide personalized recommendations that are relevant and engaging to each individual user, while also taking into account the popularity and quality of the movies. The goal is to increase user engagement and satisfaction with the movie streaming platform by providing a more personalized and enjoyable experience.

Link : https://d-nb.info/1147681678/34

#### Complexity involved :-

Recommender systems can be complex due to several reasons, including the following:

Data complexity: Recommender systems typically rely on large amounts of data, which can be challenging to manage and process. The data may come from multiple sources, including user profiles, item descriptions, and past interactions, and may need to be preprocessed and cleaned to be useful.

Algorithmic complexity: There are many different algorithms that can be used in recommender systems, and choosing the right algorithm for a particular use case can be challenging. Additionally, some algorithms can be computationally expensive, making them difficult to scale to large datasets.

Cold-start problem: Recommender systems can struggle when there is not enough data available about a user or item, which is known as the cold-start problem. This can make it challenging to provide accurate recommendations to new users or for new items that have not yet been rated by users.

Diversity: Recommender systems can sometimes suffer from the "echo chamber" effect, where users are only recommended items that are similar to what they have previously interacted with. Ensuring diversity in recommendations can be challenging but is important to provide users with a broader range of options. Evaluation: Evaluating the performance of a recommender system can be challenging, as there is no single metric that can capture all aspects of a good recommendation. Different evaluation metrics may be appropriate for different use cases, and it is important to carefully consider the trade-offs between metrics when evaluating a recommender system.

#### Project Outcome – Commercial

Recommender systems have a number of commercial outcomes for businesses that use them. Here are a few examples:

Increased sales: Recommender systems can lead to increased sales by helping customers discover products they may not have found on their own. By providing personalized recommendations based on past behavior and preferences, customers are more likely to find products that they are interested in, which can lead to more purchases.

Improved customer retention: By providing personalized recommendations, recommender systems can help businesses build stronger relationships with their customers. Customers who feel that a business understands their needs and preferences are more likely to continue shopping with that business.

Reduced marketing costs: Recommender systems can help businesses reduce their marketing costs by targeting customers with personalized recommendations instead of more expensive marketing campaigns. By recommending products that customers are likely to be interested in, businesses can achieve higher conversion rates and more effective marketing campaigns.

Better inventory management: By analyzing customer behavior and preferences, recommender systems can help businesses better manage their inventory. By predicting which products will be popular with customers, businesses can optimize their inventory levels and reduce the likelihood of overstocking or understocking.

#### 2.3.3 Project Outcome – Social ;

Recommender systems can have a number of social outcomes that can have both positive and negative impacts. Here are a few examples:

Filter bubbles: Recommender systems can contribute to the creation of filter bubbles, where users are only exposed to content that reinforces their existing beliefs and preferences. This can lead to polarization and the spread of misinformation. However, if done correctly, recommender systems can also expose users to diverse perspectives and help break down filter bubbles.

Serendipity: Recommender systems can help users discover new and interesting content that they may not have otherwise found. By presenting users with personalized recommendations based on their interests and past behavior, recommender systems can help users discover new perspectives and ideas.

Privacy: Recommender systems rely on collecting and analyzing large amounts of user data to provide personalized recommendations. This can raise concerns about user privacy and data security. It is important for businesses to be transparent about their data collection practices and to provide users with control over their data.

Accessibility: Recommender systems can help make content more accessible to users with different preferences and needs. By recommending content that is tailored to the user's interests and past behavior, recommender systems can help users navigate complex content ecosystems and find content that is relevant to their needs.

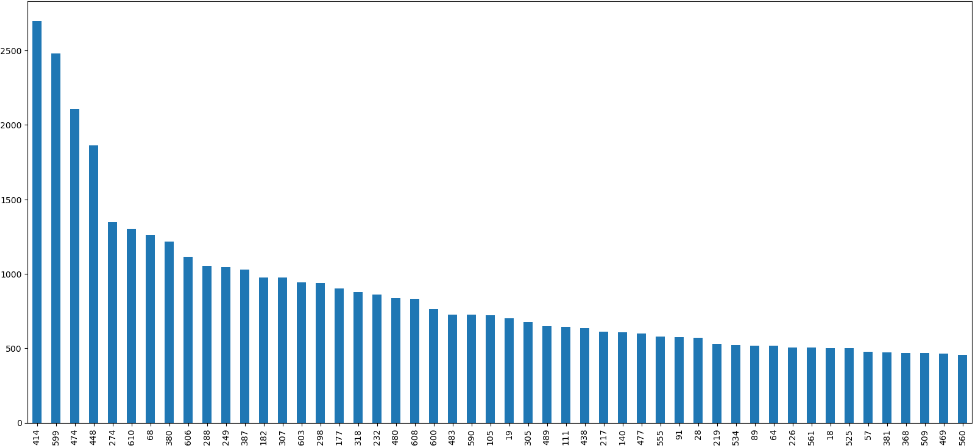
1. **Data Exploration (EDA) :-**

### Univariate Analysis:

There are a total of 6 features.

### Count of UserID column :

In our UserId there are 610 unique Id are present , we will see for top 50 .

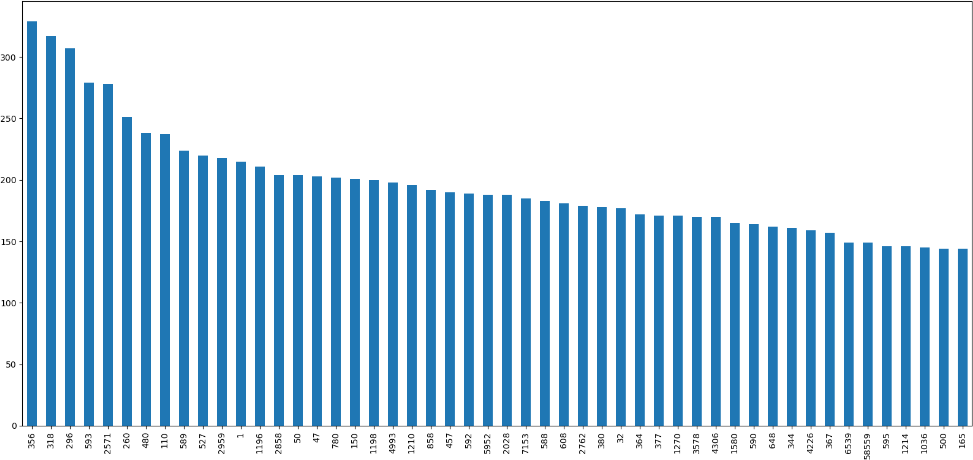


**Infrence :-**

We can see that UserID 414 has watches highesht movies and gave highest ratings in the data set.

### Count of MovieID Column:

We have 9172 unique MovieID in the dataset .

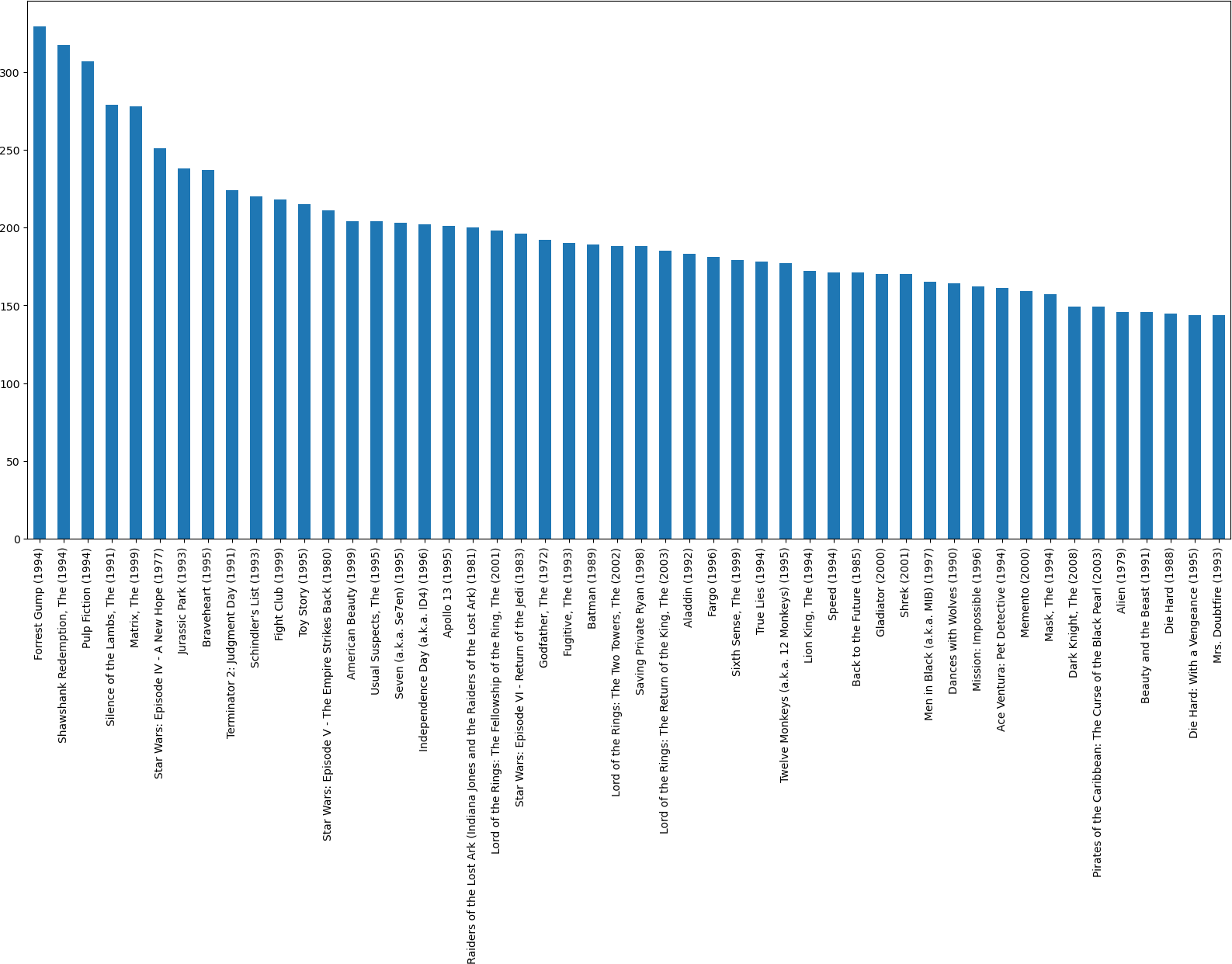


**Infrence :-**

We can see that MovieID 356 has being watched highest times in the dataset .

### Count of Movie Title Column:

We have 9172 unique MovieID in the dataset .

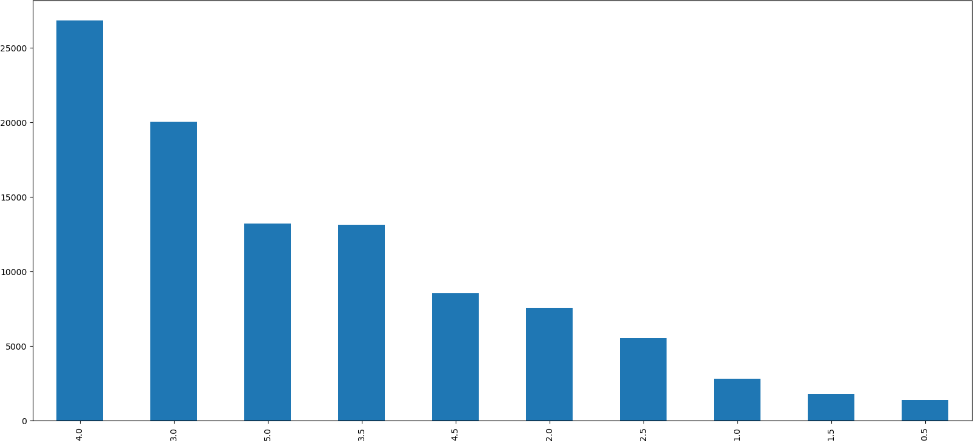


**Infrence :-**

We can see that Movie Title Forest Gump (1994) is the most watched movie in the dataset.

### Count of Rating Column:

We have 10 unique Rating given by user in the dataset .

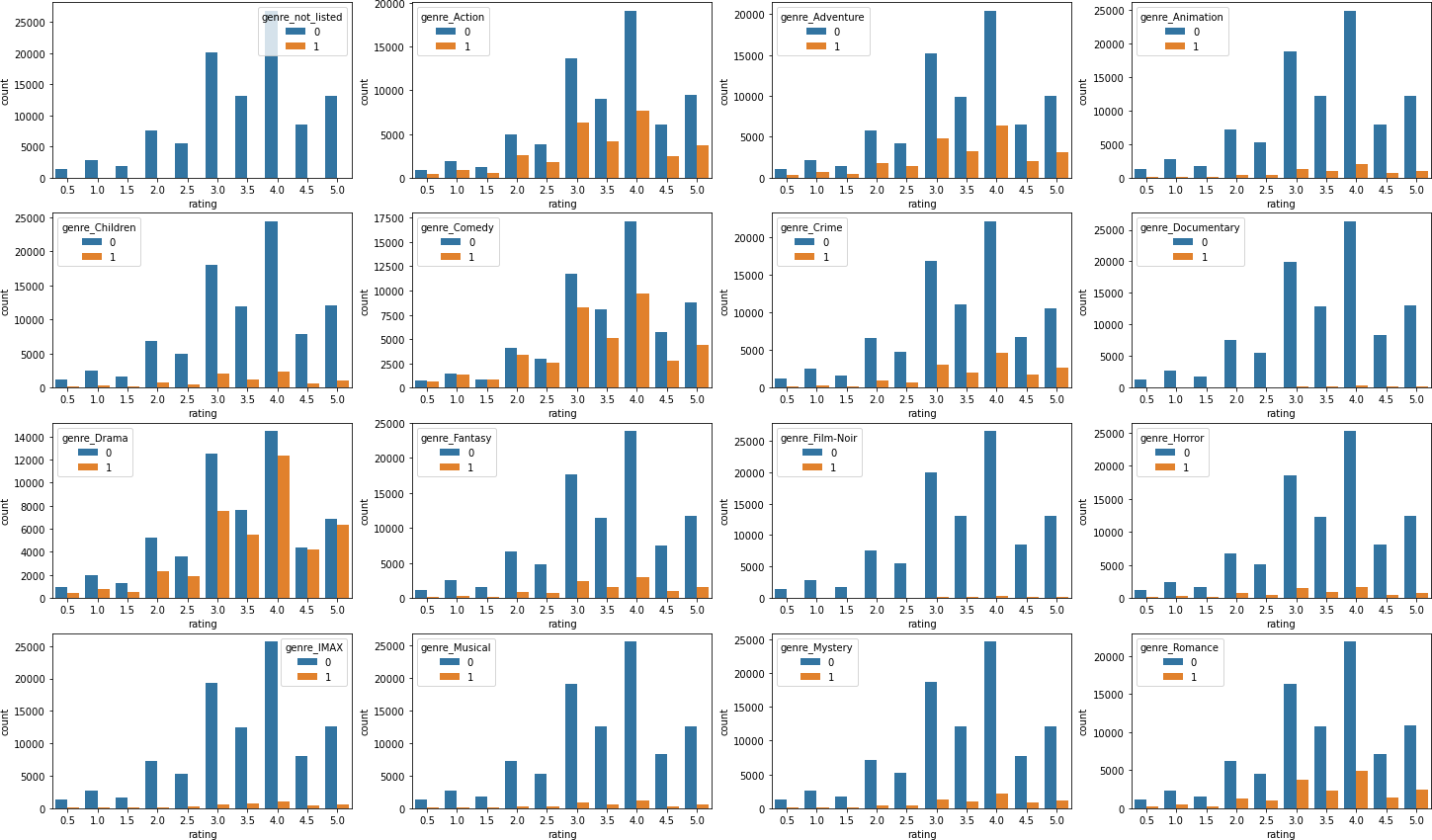


**Infrence :-**

We can see that Rating 4 is the most given rating by the users in the dataset.

### Bivariate Analysis:

Bivariate analysis between different genres and ratings.



**Inference:**

From the graph, it can be inferred that movies with drama genre has been watched most thus rated most. Drama genre as most 5.0 ratings amongst all the genres followed by comedy, action, adventure.

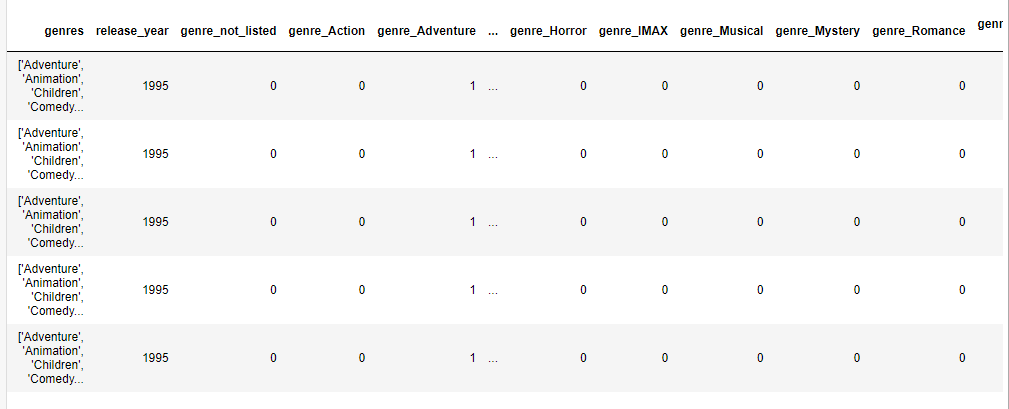
## Feature Engineering:

* 1. Our Data doesn’t need any kind of Transformation or Scaling. All the features are independent of each other.
  2. There are different genres present in genres columns in list format so we separate them and create columns for distinct genres and perform encoding on that columns.

**Before Encoding:**



**After Encoding :-**

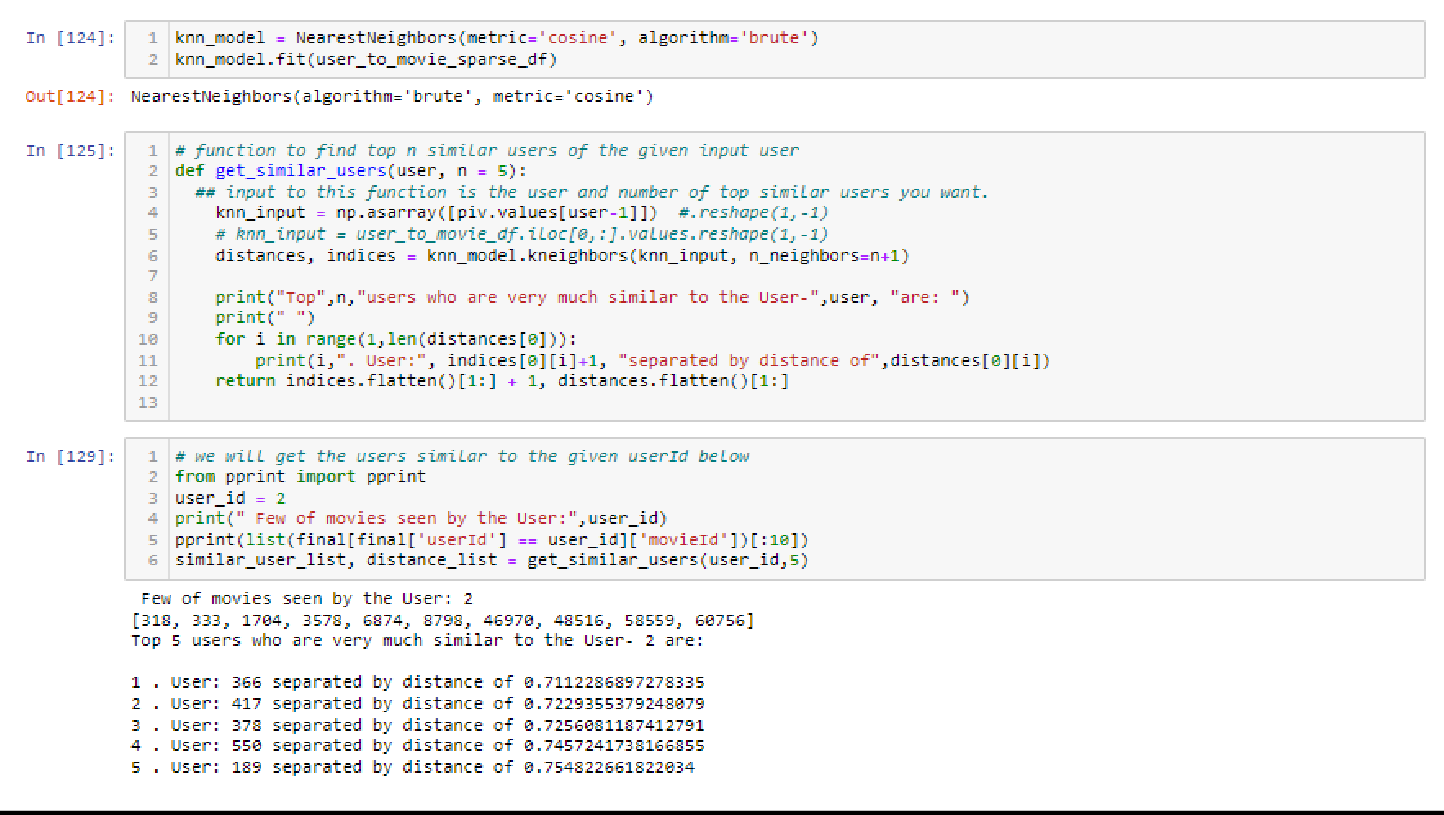


1. **Base Model :-**

-- Movie Recommendation using KNN with Input as User id, Number of similar users should the model pick and Number of movies you want to get recommended:

-- Reshaping model in such a way that each user has n- dimensional rating space where n is total number of movies

-- We will train the KNN model inorder to find the closely matching similar users to the user we give as input and we recommend the top movies which would interest the input user.

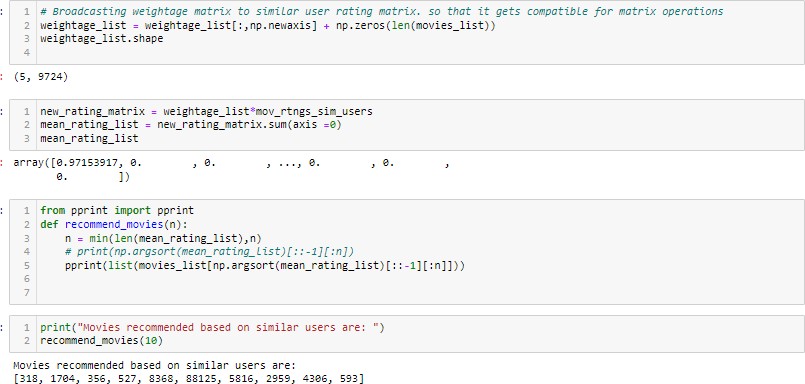


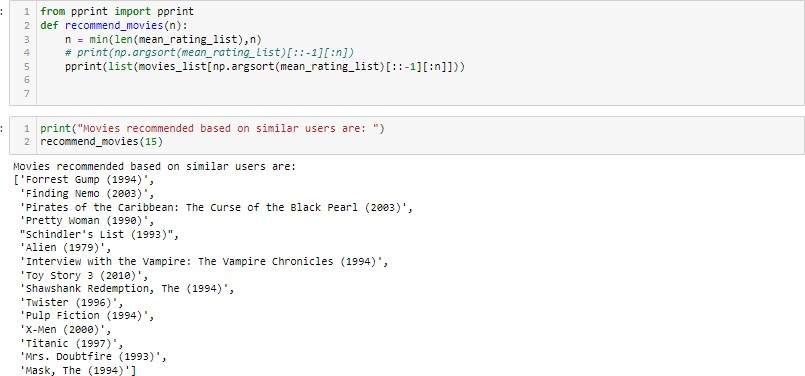
-- With the help of the KNN model built, we could get desired number of top similar users.

-- Now we will have to pick the top movies to recommend.

-- One way would be by taking the average of the existing ratings given by the similar users and picking the top 10 or 15 movies to recommend to our current user.

-- But I feel recommendation would be more effective if we define weights to ratings by each similar user based on the thier distance from the input user. Defining these weights would give us the accurate recommendations by eliminating the chance of decision manipulation by the users who are relatively very far from the input user.¶





it had been observed that, this recommendation system built can be made more efficient as it has few drawbacks.

**Drawbacks:**

1. But this recommendation system has a drawback, it also recommends movies which are already seen by the given input User.¶
2. And also there is a possibility of recommending the movies which are not at all seen by any of the similar users.

3. For new users , on what basis it recommends movie .

# 5.1 To Overcome the drawbacks :

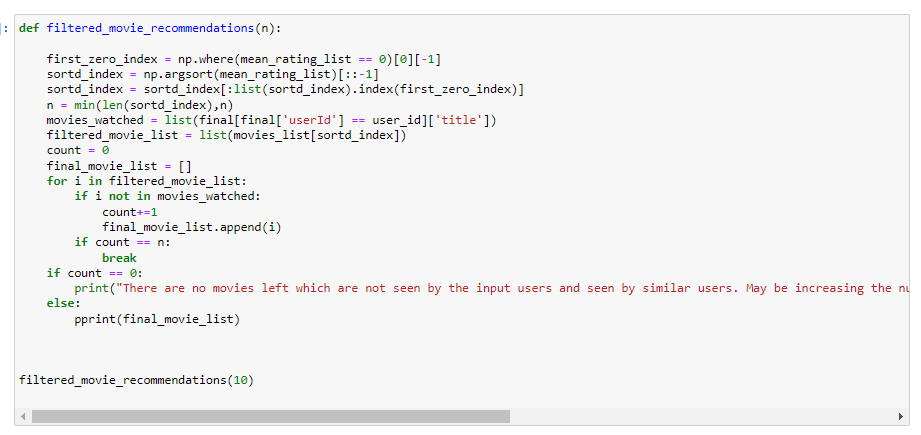
1. To address the above drawbacks we have created a new

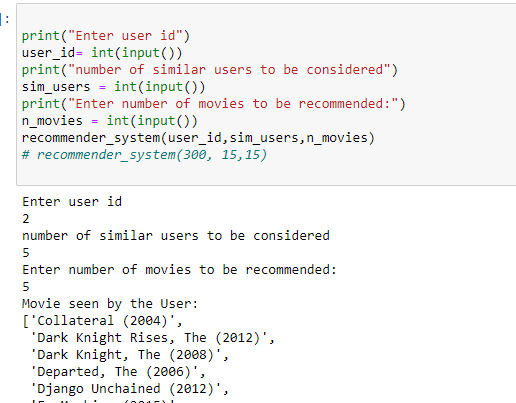
recommendation system which is a modified version of a base model .

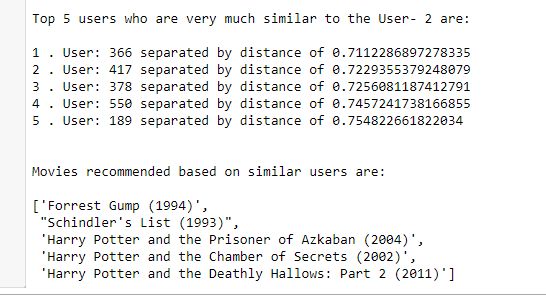
2. It recommends movies with on the basis of similarity of users but It

address the drawback that movie is not watched by user .

3. It does not recommend the movie which are not watched a single time .







-- Above are the final recommendations made by the system.

**6. other models to make recommendation system more efficient:**

**6.1 Movie Recommendation using KNN with Input as Movie Name and Number of movies you want to get recommended:**

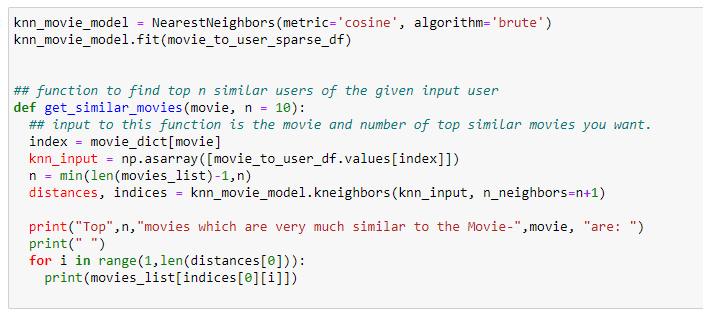
-- Reshaping model in such a way that each movie has n-dimensional

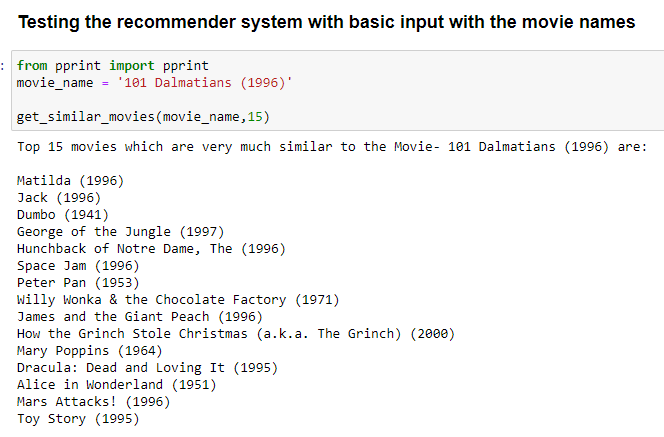
Rating space where n is total number of users who could rate.

-- We will train the KNN model in order to find the closely matching similar

movies to the movie we give as input and we recommend the top movies

which would more closely align to the movie we have given.



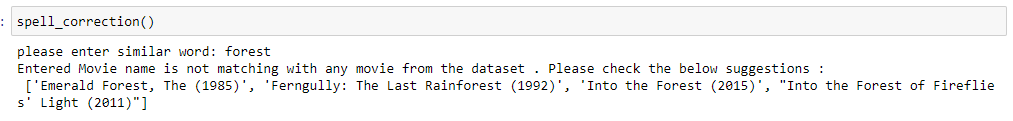


-- Now what if the user don’t know the exact movie name we have created a

Function named spell\_correction which takes few words of the movies and suggest the similar movie names.

Below is the snippet on the code and output when user searches for similar movies based on word ‘forest’:





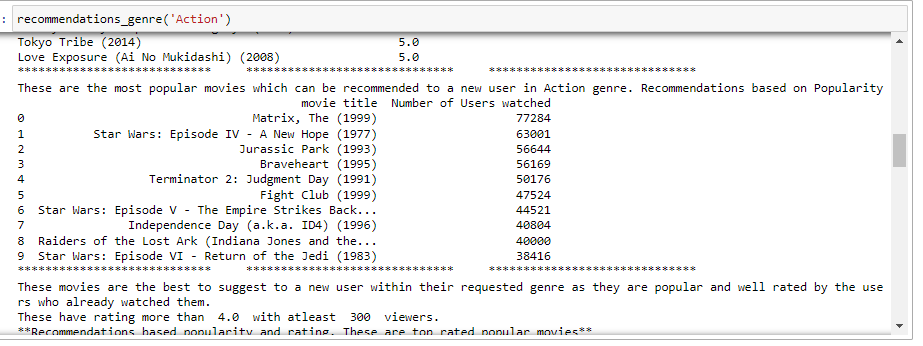
**6.2 Knowledge based recommender system :**

-- This model helps us to recommends movies to new users on the basis of genre.

-- This movie recommends top rated movies, most viewed movies and the movies whose rating is greater than 4 and view greater than 300 on the basis of genre for new users.

Here is the snippet of the Recommender system based on genre and the output of recommendation on ‘Action’ genre:





**6.3 Hybrid Recommender system** :

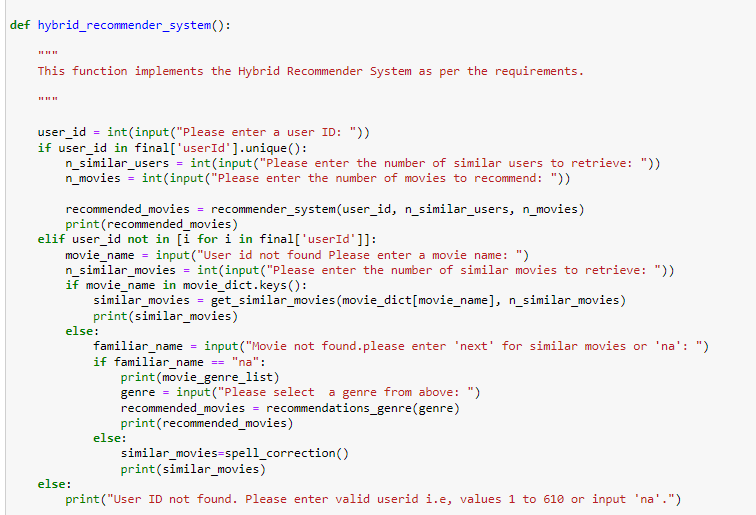
-- A hybrid recommender system is a type of recommendation system that combines two or more different recommendation techniques in order to provide more accurate and diverse recommendations. The goal of a hybrid recommender system is to overcome the limitations of individual recommendation techniques by leveraging their strengths and compensating for their weaknesses.

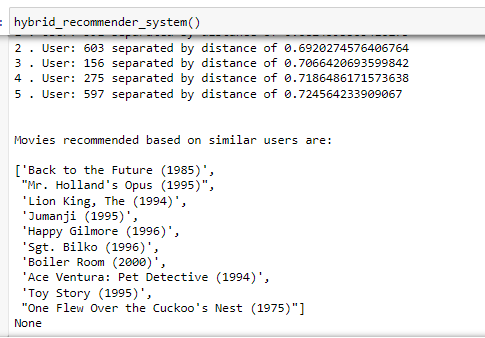
-- There are various approaches to building a hybrid recommender system. One common approach is to combine the results of different recommendation techniques using a weighted or switch-based approach. For example, the system may use collaborative filtering to recommend items that are similar to those that the user has already shown an interest in, and content-based filtering to recommend items that match the user's specific preferences.

-- Another approach is to use a cascading or mixed method, where one recommendation technique is used to filter the initial set of recommendations generated by another technique. For example, the system may use collaborative filtering to generate a set of initial recommendations, and then use content-based filtering to further refine the recommendations based on the user's specific interests.

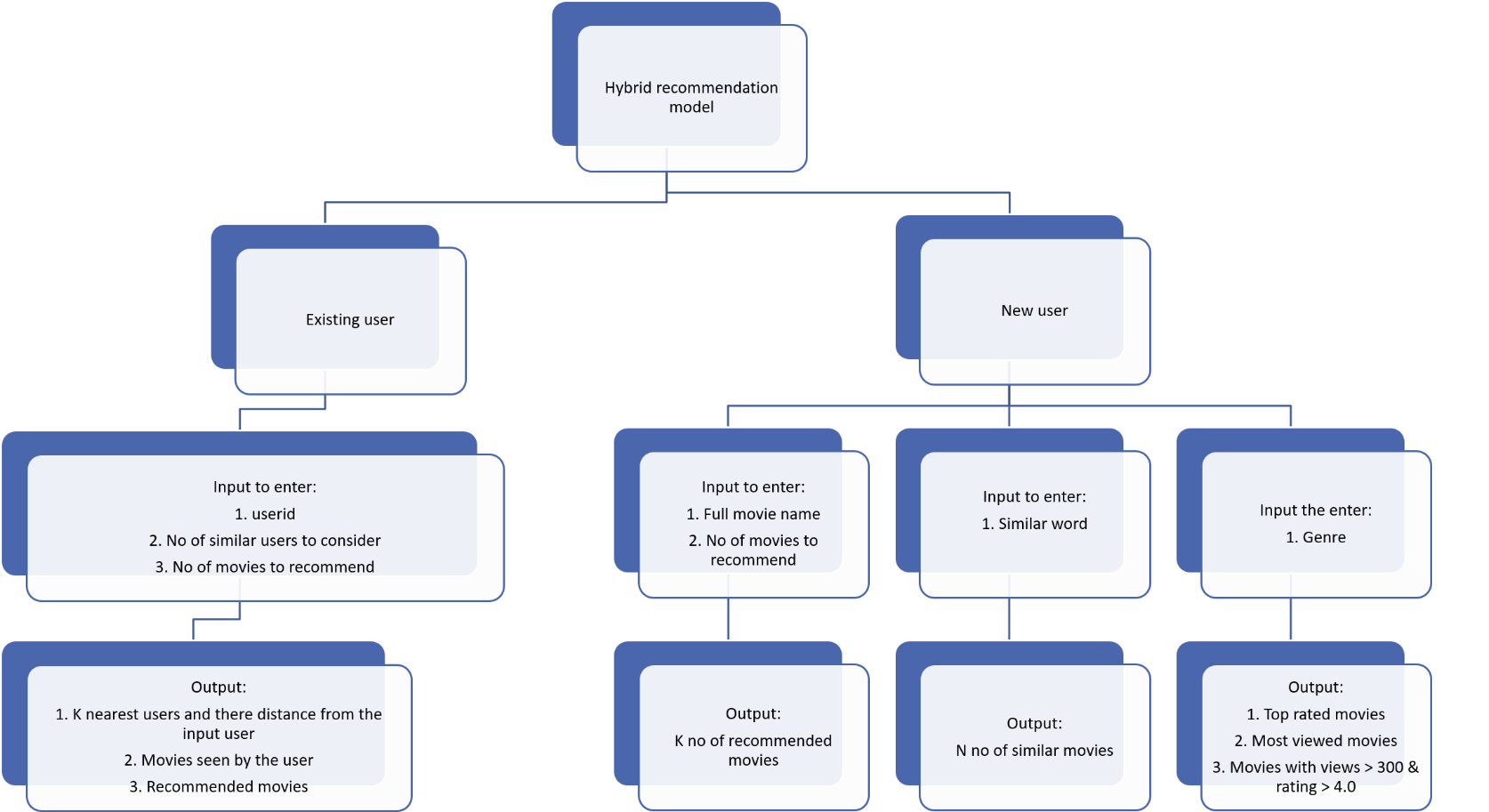
* We used switch-based approach. This function is the combination of all functions that implements a hybrid recommender system that combines two different approaches to generate personalized movie recommendations.
* First, if the user provides a valid user ID, the function calls the **recommender\_system** function to generate a list of recommended movies based on collaborative filtering.
* If the user provides an invalid user ID, the function offers the user the option to search for similar movies based on the exact movie name. If a movie name is found, the function calls the **get\_similar\_movies** function to generate a list of recommended movies based on item-item similarity.
* If the user inputs invalid movie name then the function offers the user another option to search similar movies based on familiar words. If the user inputs familiar word, the function calls the spell\_correction() function to generate a list of movies that are familiar to the input word.
* If a user doesn’t have a familiar word to search with then the function offers the user to get recommendation based on genre. The function display a list to genres available to select from. As the user inputs the genre the function call the recommendations\_genre() which generates top rated movies, most viewed movies, movies with > 300 views and rating > 4.0 of the input genre.

Below is the snippet of the code and its output when user has userID:





**Hybrid Recommendation model flow** :



----------------------- Thank You ---------------------------