greatlearning

CAPSTONE PROJECT

Movies Recommender System

GROUP-2

-Interim Report

Mentored by:

Mr. Chandran Venkateshan

Submitted by:

Mohammad Kareem Khan

Rishab Jain

Chayan Dwivedi

Deepshree Kondra

Kadakuzhill Job Himalaya

Sai Kumar

Table of Contents:

1. Industry Review :-

1. 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.

2. 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.

3. 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.

2. 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.

3. 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.

2. Dataset and Domain :-

2.1 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

2.2 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.

2.3 Project Justification :-

2.3.1 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

2.3.2 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.

2.3.3 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.

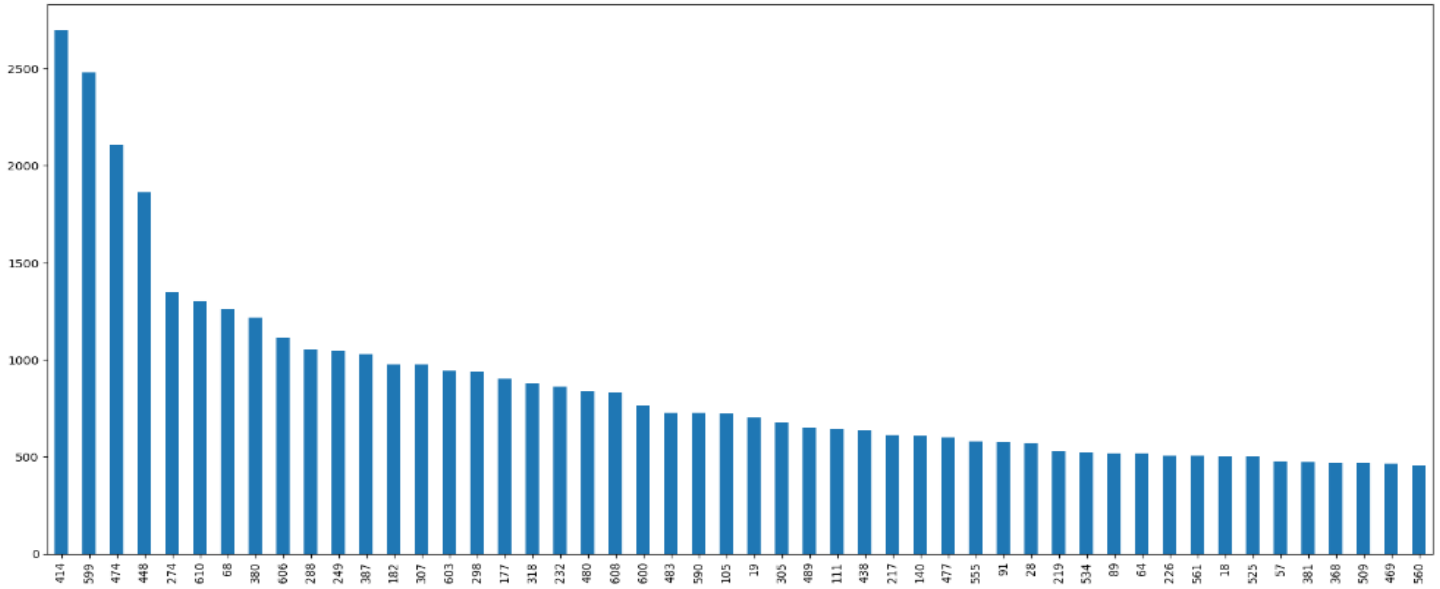
3 Data Exploration (EDA) :-

3.1 Univariate Analysis:

There are a total of 6 features.

3.1.1 Count of UserID column :

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

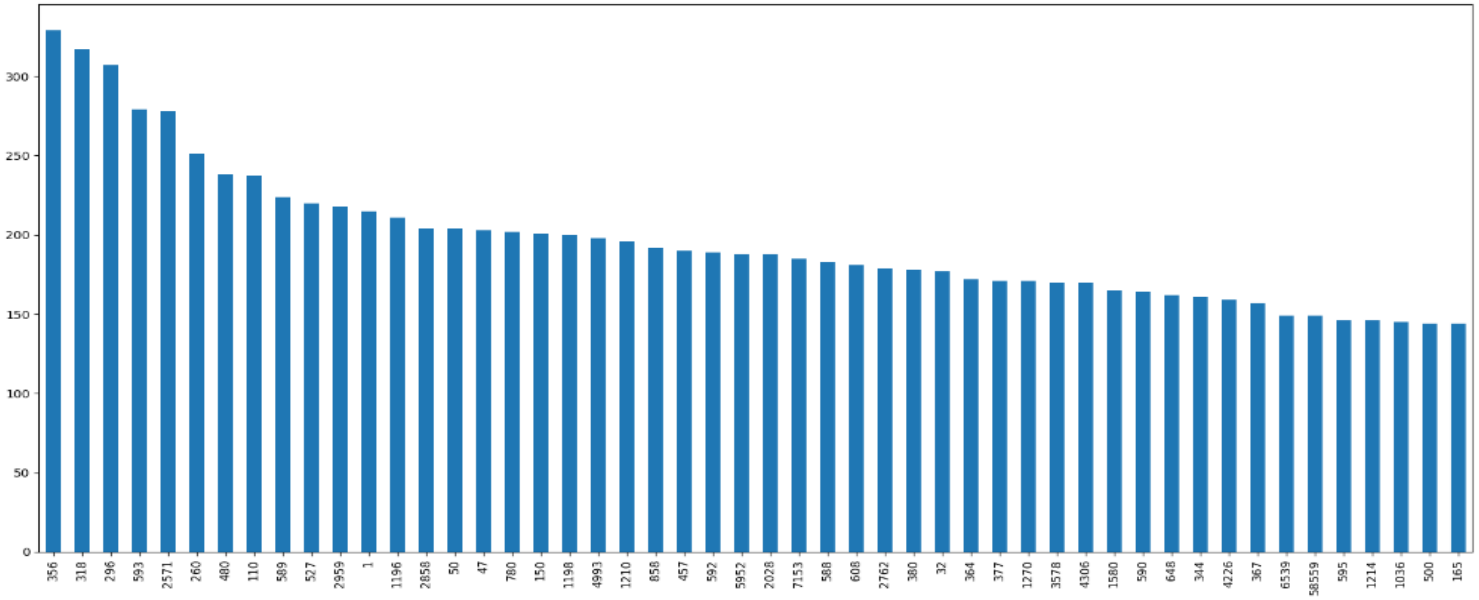


Infrence :-

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

3.1.2 Count of MovieID Column:

We have 9172 unique MovieID in the dataset .

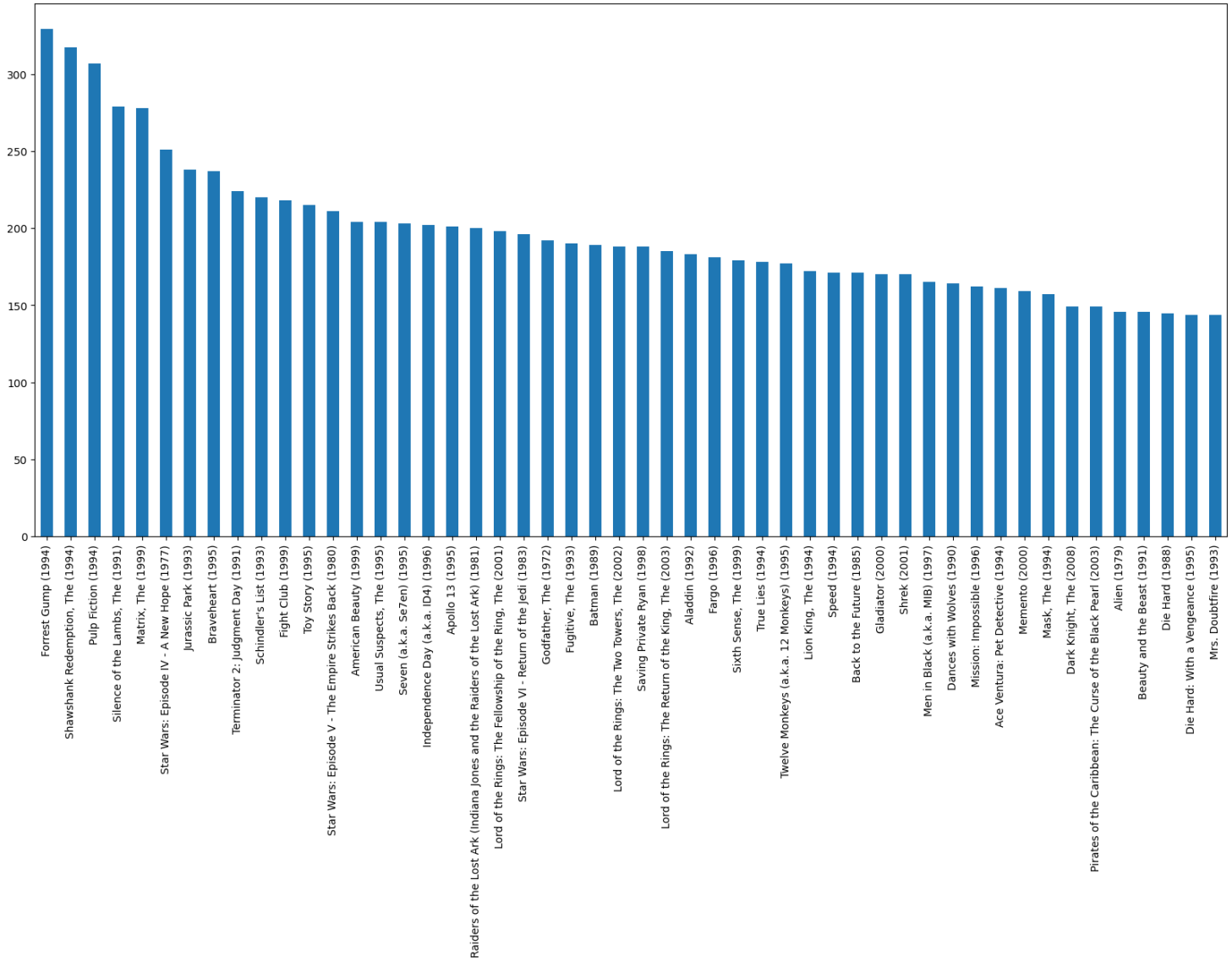


Infrence :-

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

3.1.3 Count of Movie Title Column:

We have 9172 unique MovieID in the dataset .

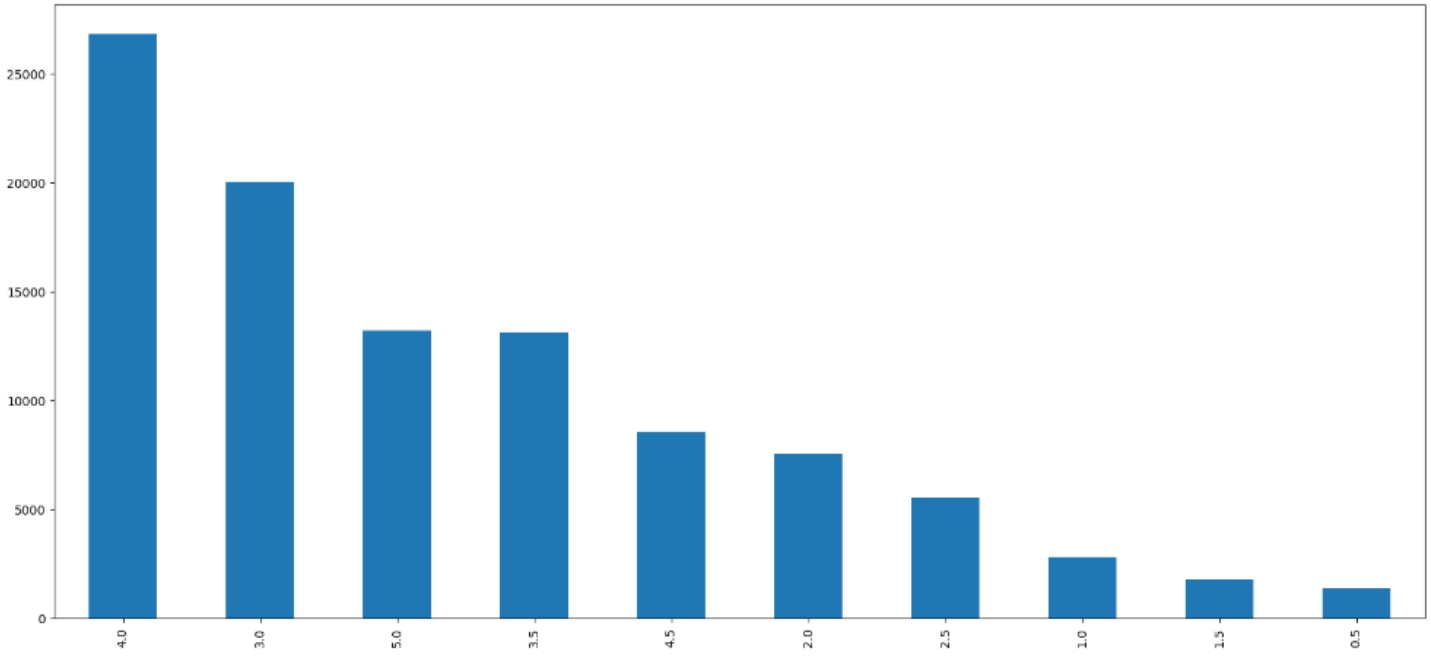


Infrence :-

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

3.1.4 Count of Rating Column:

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

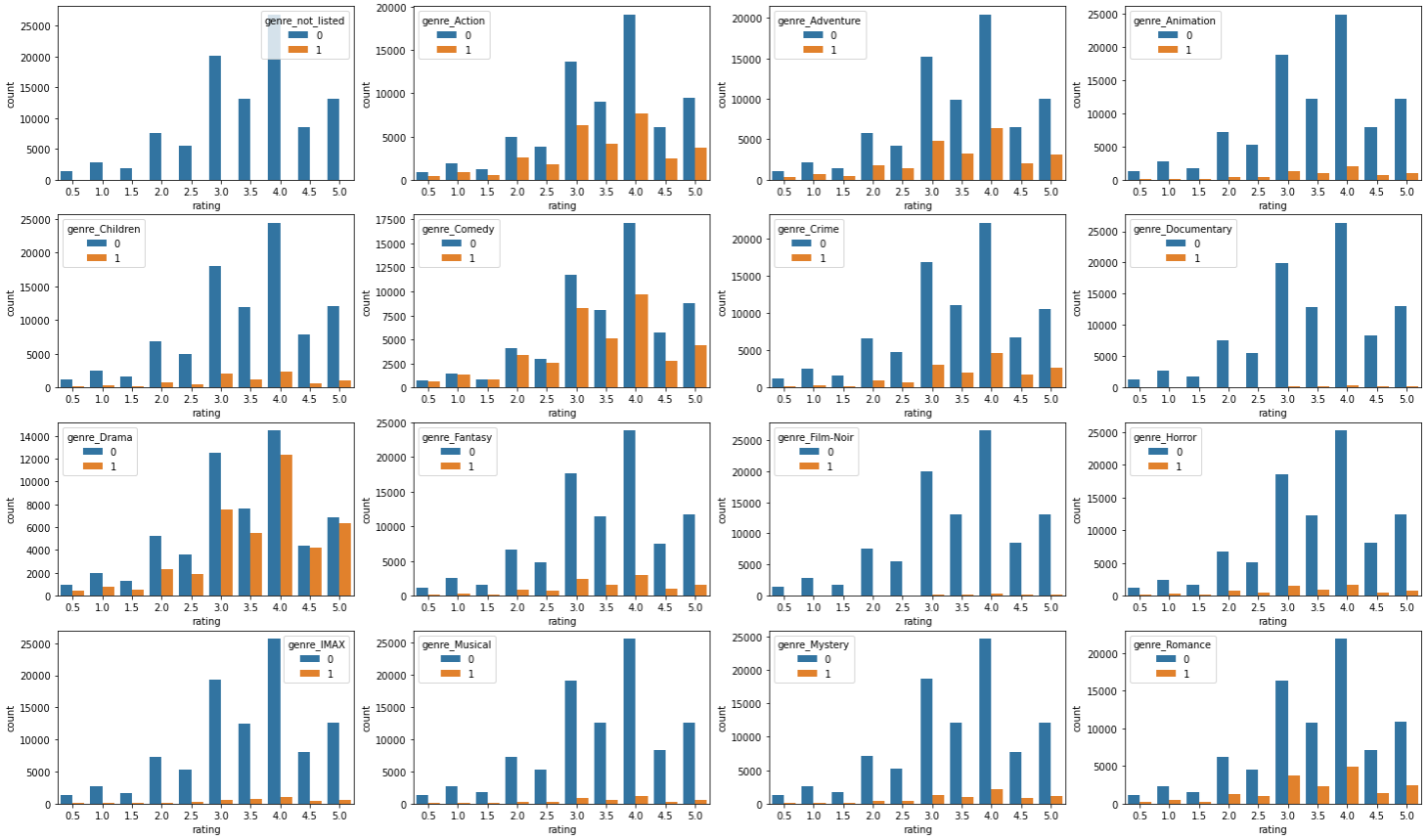


Infrence :-

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

3.2 Bivariate Analysis:

Bivatiate analysis between different genres and ratings.



Infrence :-

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 amons all the genres followed by comedy, action, adventure.

4.Feature Engineering :-

(i) Our Data doesn't need any kind of Transformation or Scaling. All the features are independent of each other .
(ii) There are different genres present in genres columns in list format so we separate them and create columns for distinct genres and perfom encoding on that columns .

Before Encoding :-

	userId	movieId	rating	timestamp	title	genres	release_year
0	1	1	4.0	964982703	Toy Story (1995)	[Adventure, Animation, Children, Comedy, Fantasy]	1995
1	5	1	4.0	847434962	Toy Story (1995)	[Adventure, Animation, Children, Comedy, Fantasy]	1995
2	7	1	4.5	1106635946	Toy Story (1995)	[Adventure, Animation, Children, Comedy, Fantasy]	1995
3	15	1	2.5	1510577970	Toy Story (1995)	[Adventure, Animation, Children, Comedy, Fantasy]	1995
4	17	1	4.5	1305696483	Toy Story (1995)	[Adventure, Animation, Children, Comedy, Fantasy]	1995

After Encoding :-

genres	release_year	genre_not_listed	genre_Action	genre_Adventure	...	genre_Horror	genre_IMAX	genre_Musical	genre_Mystery	genre_Romance	genr
['Adventure', 'Animation', 'Children', 'Comedy...	1995	0	0	1	...	0	0	0	0	0	
['Adventure', 'Animation', 'Children', 'Comedy...	1995	0	0	1	...	0	0	0	0	0	
['Adventure', 'Animation', 'Children', 'Comedy...	1995	0	0	1	...	0	0	0	0	0	
['Adventure', 'Animation', 'Children', 'Comedy...	1995	0	0	1	...	0	0	0	0	0	
['Adventure', 'Animation', 'Children', 'Comedy...	1995	0	0	1	...	0	0	0	0	0	

5.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.

In [124]:
```
knn_model = NearestNeighbors(metric='cosine', algorithm='brute')
knn_model.fit(user_to_movie_sparse_df)
```

Out[124]:
```
NearestNeighbors(algorithm='brute', metric='cosine')
```

In [125]:
```
# function to find top n similar users of the given input user
def get_similar_users(user, n = 5):
  ## input to this function is the user and number of top similar users you want.
    knn_input = np.asarray([piv.values[user-1]])  #.reshape(1,-1)
    # knn_input = user_to_movie_df.iloc[0,:].values.reshape(1,-1)
    distances, indices = knn_model.kneighbors(knn_input, n_neighbors=n+1)

    print("Top",n,"users who are very much similar to the User-",user, "are: ")
    print(" ")
    for i in range(1,len(distances[0])):
        print(i,". User:", indices[0][i]+1, "separated by distance of",distances[0][i])
    return indices.flatten()[1:] + 1, distances.flatten()[1:]

```

In [129]:
```
# we will get the users similar to the given userId below
from pprint import pprint
user_id = 2
print(" Few of movies seen by the User:",user_id)
pprint(list(final[final['userId'] == user_id]['movieId'])[:10])
similar_user_list, distance_list = get_similar_users(user_id,5)
```

```
 Few of movies seen by the User: 2
[318, 333, 1704, 3578, 6874, 8798, 46970, 48516, 58559, 60756]
Top 5 users who are very much similar to the User- 2 are:

1 . User: 366 separated by distance of 0.7112286897278335
2 . User: 417 separated by distance of 0.7229355379248079
3 . User: 378 separated by distance of 0.7256081187412791
4 . User: 550 separated by distance of 0.7457241738166855
5 . User: 189 separated by distance of 0.754822661822034
```

-- 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.¶

```
# Broadcasting weightage matrix to similar user rating matrix. so that it gets compatible for matrix operations
weightage_list = weightage_list[:,np.newaxis] + np.zeros(len(movies_list))
weightage_list.shape

```

```
(5, 9724)
```

```
new_rating_matrix = weightage_list*mov_rtngs_sim_users
mean_rating_list = new_rating_matrix.sum(axis =0)
mean_rating_list
```

```
array([0.97153917, 0.        , 0.        , ..., 0.        , 0.        ,
       0.        ])
```

```
from pprint import pprint
def recommend_movies(n):
    n = min(len(mean_rating_list),n)
    # print(np.argsort(mean_rating_list)[::-1][:n])
    pprint(list(movies_list[np.argsort(mean_rating_list)[::-1][:n]]))


```

```
print("Movies recommended based on similar users are: ")
recommend_movies(10)
```

```
Movies recommended based on similar users are:
[318, 1704, 356, 527, 8368, 88125, 5816, 2959, 4306, 593]
```

```
from pprint import pprint
def recommend_movies(n):
    n = min(len(mean_rating_list),n)
    # print(np.argsort(mean_rating_list)[::-1][:n])
    pprint(list(movies_list[np.argsort(mean_rating_list)[::-1][:n]]))
```

```
print("Movies recommended based on similar users are: ")
recommend_movies(15)
```

```
Movies recommended based on similar users are:
['Forrest Gump (1994)',
 'Finding Nemo (2003)',
 'Pirates of the Caribbean: The Curse of the Black Pearl (2003)',
 'Pretty Woman (1990)',
 "Schindler's List (1993)",
 'Alien (1979)',
 'Interview with the Vampire: The Vampire Chronicles (1994)',
 'Toy Story 3 (2010)',
 'Shawshank Redemption, The (1994)',
 'Twister (1996)',
 'Pulp Fiction (1994)',
 'X-Men (2000)',
 'Titanic (1997)',
 'Mrs. Doubtfire (1993)',
 'Mask, The (1994)']
```

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.

5.1 Future Work

-- Above drawbacks are addressed and a new recommender system with modification is in progress .

-- As we built the model on user-based recommender system further we will try to built a content-based recommender system if possible , we will built hybrid recommender system which is combination of user-based and content-based recommender system .

-- Now we have only built our base model we try to modify the model to get better recommendations .