INNOMATICS RESEARCH LABS

Hyderabad, Telangana 500085

"MOVIE RECOMMENDATION SYSTEM"

Submitted in Partial fulfillment of the Requirements for the

Internship Program

By

YASHSHREE BAVISKAR

RAVI TEJA CH

NILKANTHA BAG

SRINEHA G

CHAITANYA DUBAL

Under the Guidance of,

KOUSTHUBH PANYALA

INNOMATICS RESEARCH LABS

 # 205, Fortune Signature, beside JNTU Metro Station, Sardar Patel Nagar, IDPL Staff Cooperative Housing Society, Kukatpally Housing Board Colony, Kukatpally, Hyderabad, Telangana 500085

DECLARATION

We, the students of Innomatics Research labs declare that the work entitled "**MOVIE RECOMMENDATION SYSTEM**" has been successfully completed under the guidance of **Kousthubh Panyala**, this dissertation work is submitted in partial fulfillment of the requirements for the award certificate

Place:

Date:

Team members:

Yashshree Baviskar

Ravi Teja CH

Nilkantha Bag

Srineha G

Chaitanya Dubal

ABSTRACT

In this hustling world, entertainment is a necessity for each one of us to refresh our mood and energy. Entertainment regains our confidence for work and we can work more enthusiastically. For revitalizing ourselves, we can listen to our preferred music or can watch movies of our choice. For watching favorable movies online, we can utilize movie recommendation systems, which are more reliable, since searching of preferred movies will require more and more time which one cannot afford to waste. In this report, to improve the quality of a movie recommendation system, a Hybrid approach by combining content based filtering and collaborative filtering, using Support Vector Machine as a classifier and genetic algorithm is presented in the proposed methodology and comparative results have been shown which depicts shows an improvement in the accuracy, quality, and scalability of the movie recommendation system than the pure approaches in three different datasets. Hybrid approach helps to get the advantages from both the approaches as well as tries to eliminate the drawbacks of both methods.

TABLE OF CONTENTS

Page No.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Relevance of the Project

A recommendation system or recommendation engine is a model used for information filtering where it tries to predict the preferences of a user and provide suggests based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places, and other utilities. These systems collect information about a user's preferences and behavior, and then use this information to improve their suggestions in the future.

Movies are a part and parcel of life. There are different types of movies like some for entertainment, some for educational purposes, some are animated movies for children, and some are horror movies or action films. Movies can be easily differentiated through their genres like comedy, thriller, animation, action etc. Other way to distinguish among movies can be either by releasing year, language, director etc. Watching movies online, there are several movies to search in our most liked movies. Movie Recommendation Systems helps us to search our preferred movies among all these different types of movies and hence reduce the trouble of spending a lot of time searching our favorable movies. So, it requires that the movie recommendation system should be very reliable and should provide us with the recommendation of movies which are exactly same or most matched with our preferences.

Many companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffer with poor recommendation quality and scalability issues.

1.2 Problem Statement:

The goal of the project is to recommend a movie to the user.

Providing related content out of relevant and irrelevant collection of items to users of online service providers.

1.3 Objective of the Projects

- Improving the Accuracy of the recommendation system
- Improve the Quality of the movie Recommendation system
- Improving the Scalability.
- Enhancing the user experience.

1.4 Scope of the Project

The objective of this project is to provide accurate movie recommendations to users. The goal of the project is to improve the quality of movie recommendation system, such as accuracy, quality, and scalability of system than the pure approaches. This is done using Hybrid approach by combining content based filtering and collaborative filtering, to eradicate the overload of the data, recommendation system is used as information filtering tool in social networking sites. Hence, there is a huge scope of exploration in this field for improving scalability, accuracy and quality of movie recommendation systems Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffer with poor recommendation quality and scalability issues.

1.5 Methodology for Movie Recommendation

The hybrid approach proposed an integrative method by merging fuzzy k- means clustering method and genetic algorithm based weighted similaritymeasure to construct a movie recommendation system. The proposed movie recommendation system gives finer similarity metrics and quality than the existing Movie recommendation system but the computation time which is

taken by the proposed recommendation system is more than the existingrecommendation system. This problem can be fixed by taking the clustered data points as an input dataset

The proposed approach is for improving the scalability and quality of the movie recommendation system. We use a Hybrid approach, by unifying Content-Based Filtering and Collaborative Filtering, so that the approachescan be profited from each other. For computing similarity between the different movies in the given dataset efficiently and in least time and to reducecomputation time of the movie recommender engine we used cosine similaritymeasure.

Agile Methodology:

1.   collecting the data sets: Collecting all the required data set from Kaggle web site.in this project we require movie.csv, ratings.csv, users.csv.

2.   Data Analysis: make sure that that the collected data sets are correct and analyzing the data in the csv files. i.e., checking whether all the column Felds are present in the data sets.

3.   Algorithms: in our project we have only two algorithms one is cosine similarity and other is single valued decomposition are used to build the machine learning recommendation model.

CHAPTER 2

LITERATURE SURVEY

Over the years, many recommendation systems have been developed using either collaborative, content based or hybrid filtering methods. These systems have been implemented using various big data and machine learning algorithms.

2.1. Movie Recommendation System by K-Means ClusteringAnd K-Nearest Neighbor

A recommendation system collect data about the user's preferences either implicitly or explicitly on different items like movies. An implicit acquisition in the development of movie recommendation system uses the user's behaviour while watching the movies. On the other hand, a explicit acquisition in the development of movie recommendation system uses the user's previous ratings or history. The other supporting technique that are used in the development of recommendation system is clustering. Clustering is a process to group a set of objects in such a way that objects in the same clusters are more like each other than to those in other clusters. K- Means Clustering along with K-Nearest Neighbour is implemented on the movie lens dataset in order to obtain the best-optimized result. In existing technique, the data is scattered which results in a high number of clusters while in the proposed technique data is gathered and results in a low number of clusters. The process of recommendation of a movie is optimized in the proposed scheme. The proposed recommender system predicts the user's preference of a movie on the basis of different parameters. The recommender system works on the concept that people are having common preference or choice. These users will influence on each other's opinions. This process optimizes the process and having lower RMSE.

2.1 Movie Recommendation System Using CollaborativeFiltering:

Collaborative filtering systems analyses the user's behavior and preferences and predict what they would like based on similarity with other users. There are two kinds of collaborative filtering systems; user-based recommender and item-based recommender.

1. Use-based filtering: User-based preferences are very common in the field of designing personalized systems. This approach is based on the user's likings. The process starts with users giving ratings (1-5) to some movies. These ratings can be implicit or explicit. Explicit ratings are when the user explicitly rates the item on some scale or indicates a thumbs-up/thumbs-down to the item. Often explicit ratings are hard to gather as not every user is much interested in providing feedbacks. In these scenarios, we gather implicit ratings based on their behavior. For instance, if a user buys a product more than once, it indicates a positive preference. In context to movie systems, we can imply that if a user watches the entire movie, he/she has some likeabilityto it. Note that there are no clear rules in determining implicit ratings. Next, for each user, we first find some defined number of nearest neighbors. We calculate correlation between users' ratings using Pearson Correlationalgorithm. The assumption that if two users' ratings are highly correlated, thenthese two users must enjoy similar items and products is used to recommend items to users.

2. Item-based filtering: Unlike the user-based filtering method, item- based focuses on the similarity between the item's users like instead of the users themselves. The most similar items are computed ahead of time. Then for recommendation, the items that are most like the target item are recommended to the user.

## 2.2 Movie Recommendation System Using Popularity:

It is a type of recommendation system which works on the principle of popularity and or anything which is in trend. These systems check about the product or movie which are in trend or are most popular among the users and directly recommend those. It does not suffer from cold start problems which means on day 1 of the business also it can recommend products on various filters. There is no need for the user's historical data.

The popularity-based recommendation system eliminates the need for knowing other factors like user browsing history, user preferences, the star cast of the movie, genre, and other factors. Hence, the single-most factor considered is the star rating to generate a scalable recommendation system. This increases the chances of user engagement as compared to when there was no recommendation system.

2.3 Movie Recommendation System Using Hybrid Model:

A hybrid recommendation system is a unique subtype of recommendation system that combines collaborative and content filtering techniques. In some circumstances, combining collaborative and content-based filtering can improve performance and help us overcome the drawbacks of utilizing them independently. There are several ways to implement hybrid recommender system approaches, including employing content and collaborative-based methods to generate predictions separately, then combining the predictions, or simply enhancing a content-based approach with the capabilities of collaborative-based methods (and vice versa).

Numerous studies that assess the effectiveness of hybrid methods to conventional ways conclude that utilizing hybrid methods will result in suggestions that are more accurate.
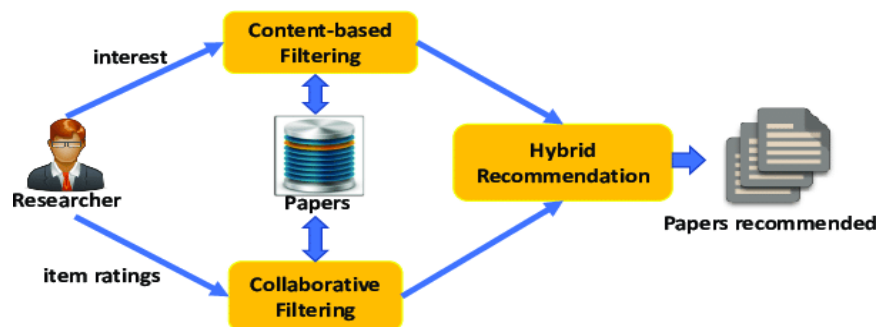
- Obtaining the Movies Metadata file and associated datasets, which include ratings and small subsets of all movies' cast and crew lists and plot keywords, is the first step. After that, we will build the **TF-IDF matrix using the TF-IDF vectorizer**, compute **cosine similarity** here, extract the relevant movie indexes, and build a recommendation model based on cosine similarity.

- We will divide the data into folds and use **Single Value Decomposition (SVD)** for cross-validation and fitting before developing a second model to create a hybrid model in order to build user-focused recommender systems.

- Finally, we will combine the two models to create a hybrid recommendation model, using tmdb-id to group related films together, rank those films based on the estimated prediction, and then show the top 15 films from that group.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

This chapter involves both the hardware and software requirements needed for theproject and detailed explanation of the specifications.

3.1 Hardware Requirements
- A PC with Windows/Linux OS
- Processor with 1.7-2.4gHz speed
- Minimum of 8gb RAM
- 2gb Graphic card

3.2 Software Specification
- Text Editor (VS-code/WebStorm)
- Anaconda distribution package (PyCharm Editor)
- Python libraries

3.3 Software Requirements

3.3.1 Anaconda distribution:

Anaconda is a free and open-source distribution of the Python programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management system and deployment. Package versions are managed by the package management system conda. The anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.3

3.3.3 Python libraries:

For the computation and analysis, we need certain python libraries which are used to perform analytics. Packages such as SKlearn, Numpy, pandas, Matplotlib, streamlit framework, etc. are needed.

SKlearn: It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working withthese arrays. It is the fundamental package for scientific computing with Python. Pandas: Pandas is one of the most widely used python libraries in datascience. It provides high-performance, easy to use structures and data analysistools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.

CHAPTER 4
Analytical Problem Framing

4.1. Mathematical / Analytical Modelling of the Problem:

Whenever we employ any ML algorithm, statistical models, or feature pre-processing in background lot of mathematical framework work. In this project we have done lot of data pre-processing & ML model building. Different ML algorithm used in this project has its own mathematical background, for which you can refer Scikit documentation [1].

4.2. Data Sources and their formats:

credits	⊘	01-10-2022 21:20	Microsoft Excel C...	1,85,467 KB
links	⊘	01-10-2022 21:19	Microsoft Excel C...	966 KB
links_small	⊘	01-10-2022 21:19	Microsoft Excel C...	180 KB
Movie_metadata	⊘	01-10-2022 21:21	Microsoft Excel C...	33,638 KB
ratings_small	⊘	01-10-2022 21:23	Microsoft Excel C...	2,382 KB

Data is collected from Kaggle.com and saved in csv files.

4.3. Loading the Datasets:

Importing the datasets by using the libraries.


```
Importing required Liabraries

import numpy as np # for numrical caluation analysis
import pandas as pd # for data wrangling
import warnings
warnings.filterwarnings('ignore')

Movies_Data=pd.read_csv("movies_metadata.csv")

print('No. of Rows:', Movies_Data.shape[0])
print('No. of Columns:',Movies_Data.shape[1])
pd.set_option('display.max_columns',None) # This will e
Movies_Data.head(2)

No. of Rows: 45466
No. of Columns: 24
```

4.4 Data Preprocessing:

The dataset is large and it may contain some data error. In order to reach clean, error free data some pre-processing is done on data. First task is to finding error in data or data entry correction. Second task is to convert datatypes in appropriate datatypes. Third one is to perform feature engineering. Different data error found in data as followed:


```
import numpy as np

df=pd.read_csv("ratings_small.csv")
df
```

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182

```
Keywords_Data=pd.read_csv("keywords.csv")
print('No. of Rows:', Keywords_Data.shape[0])
print('No. of Columns:',Keywords_Data.shape[1])
pd.set_option('display.max_columns',None) # This will e
Keywords_Data.head()

No. of Rows: 46419
No. of Columns: 2
```
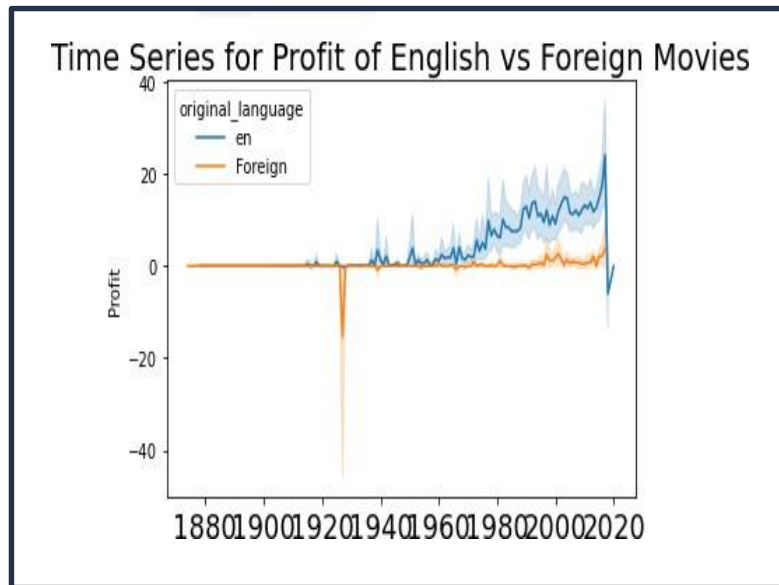
```
Credits_Data=pd.read_csv('credits.csv') # importing Cre
print('No. of Rows:', Credits_Data.shape[0])
print('No. of Columns:',Credits_Data.shape[1])
pd.set_option('display.max_columns',None) # This will e
Credits_Data.head()
```

```
No. of Rows: 45476
No. of Columns: 3
```

```python
# defining the literal_ eval function to remove the space
def fetch_director(text):
    M = []
    for i in ast.literal_eval(text):
        if i['job'] == 'Director':
            M.append(i['name'])
    return M
```

```python
import ast
```

```python
# Applying literal_eval for 'Crew' feature

Movies['crew'] = Movies['crew'].apply(fetch_director)
```

```python
Movies.head(3)
```

	id	title	overview	genres	keywords	cast	crew	budget	rev
0	862	Toy Story	Led by Woody, Andy's toys live happily in his ...	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	[{'id': 931, 'name': 'jealousy'}, {'id': 4290,...	[{'cast_id': 14, 'character': 'Woody (voice)',...	[John Lasseter]	30000000	373554

    Initially load the data sets that are required to build a model the data set that are required in this project are movies.csv, ratinfg.csv, users.csv all the data sets are available in the Kaggle.com. Basically, two models are built in this project content based and collaborative filtering each produce a list of movies to a particular user by combining both based on the user id a single final list of movies are recommended to the user.

4.5 Data Visualizations:

### 4.5.1 For Content Based Approach:

Graphs shows that year 2000-2020 profit margin is high of English movie other language or foreign movies.
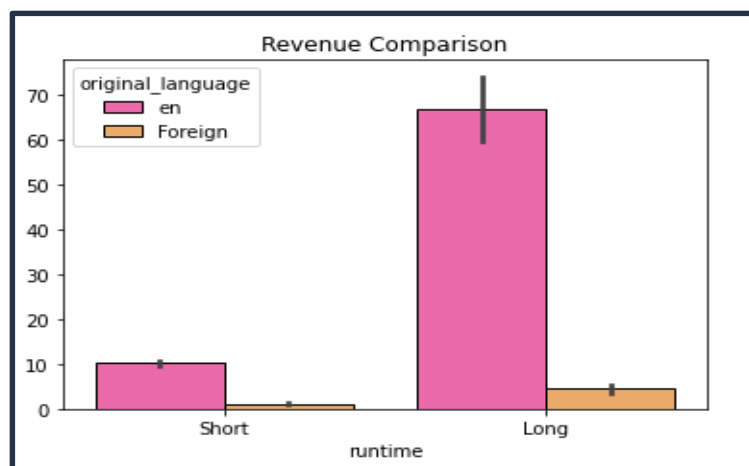
Long duration movies have more profit than short duration movies.

```python
import seaborn as sns
import matplotlib.pyplot as plt
df['tag_len'] = df['tags'].apply(lambda x: len(x))

# illustrate the tag text length
sns.displot(data=df.dropna(), bins=list(range(0, 2000, 25)), height=5, x='tag_len',
            aspect=3, kde=True)
plt.title('Distribution of tag text length')
plt.xlim([0, 2500])
```

(0.0, 2500.0)

Distribution of tag text length
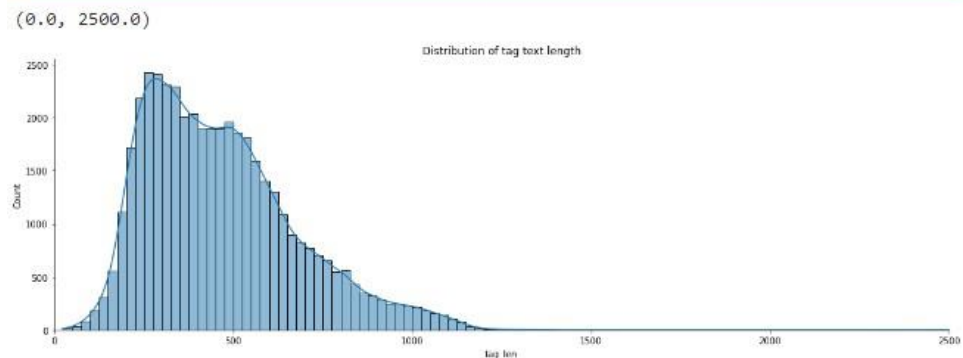
Observation: above barchat shows text length is varies between 0 t0 1250 words approximately.

4.5.2 For Collaborative Based Approach:

Movies with most reviews

```python
# univariate analysis
plt.figure(1, figsize = (16,4))
df['movieId'].value_counts()[:50].plot(kind = 'bar') #take top 50 movies
plt.figure(2, figsize = (16,4))
df['userId'].value_counts()[:50].plot(kind = 'bar') #take top 50 users
plt.figure(3, figsize = (8,4))
df['rating'].plot(kind = 'hist')
```
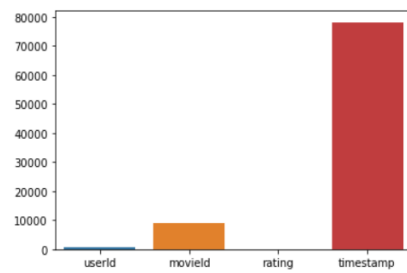
<AxesSubplot:ylabel='Frequency'>
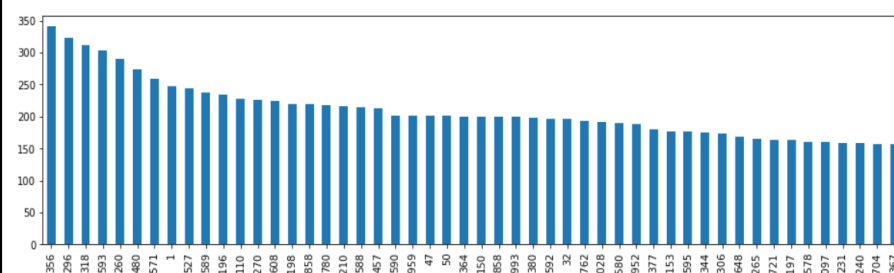
Distinct users and movies are included in the dataset

```
print(df.nunique())
```

```
userId        671
movieId      9066
rating         10
timestamp   78141
dtype: int64
```
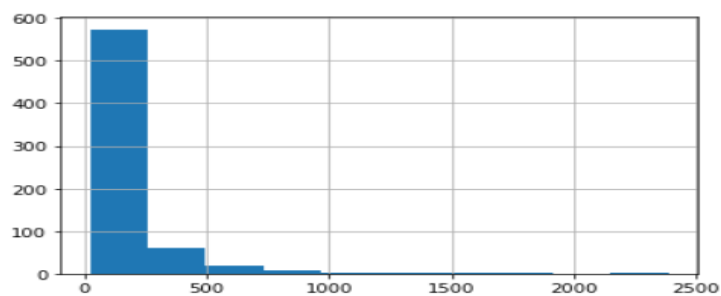
```
sns.barplot(x=df.columns,y=df.nunique())
```

```
<AxesSubplot:>
```


Ratings given to each movie

```
ratings_per_user = df.groupby('userId')['movieId'].count()
ratings_per_user.hist()
```
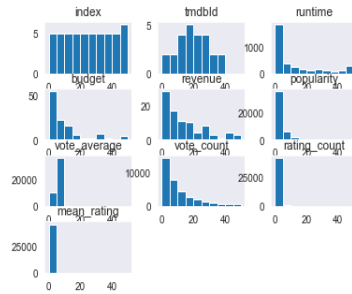
```
<AxesSubplot:>
```



```
# most users (roughly 560 out of 671 -80%) have less than 250 ratings.
```

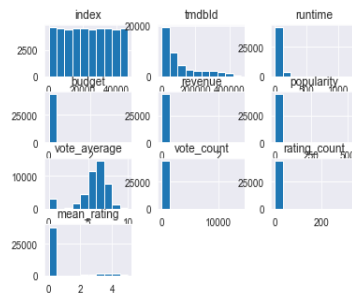## 4.5.3 For Popularity Based Approach:

```
In [76]: data.hist(grid=False, bins=range(0,55,5))
         plt.xlabel('vote_average')
         plt.ylabel('No.of votes')
         plt.title('votes')
```

Out[76]: Text(0.5, 1.0, 'votes')


```
In [79]: data.hist()
         plt.xlabel('popularity')
         plt.ylabel('rating_count')
         plt.title('popularity')
```
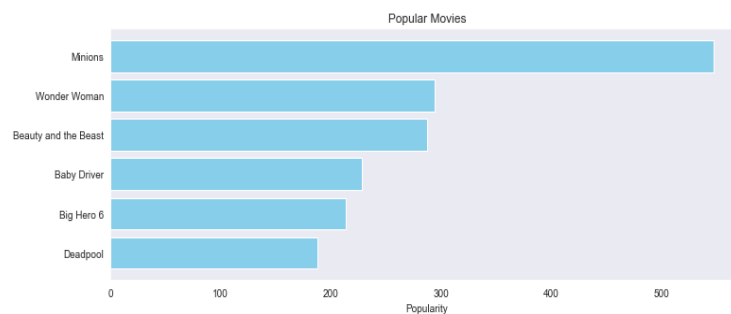
Out[79]: Text(0.5, 1.0, 'popularity')


```
In [45]: pop= data.sort_values('popularity', ascending=False)
         import matplotlib.pyplot as plt
         plt.figure(figsize=(12,4))

         plt.barh(pop['title'].head(6),pop['popularity'].head(6), align='center',
                 color='skyblue')
         plt.gca().invert_yaxis()
         plt.xlabel("Popularity")
         plt.title("Popular Movies")
```

Out[45]: Text(0.5, 1.0, 'Popular Movies')

CHAPTER 5

IMPLEMENTATION

The Proposed System Make Use Different Algorithms and Methods for the implementation of popularity based and content based, collaborative based and hybrid based Approach.

5.1 Cosine Similarity: **Cosine similarity** is a measure of similarity betweentwo non-zero vectors of an inner product space that measures the cosine of the anglebetween them.

Formula:

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \ \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \ \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

5.2 Singular Value Decomposition (SVD):

Let A be an n*d matrix with singular vectors v1, v2, . . ., vr and corresponding singular values σ1, σ2, . . . , σr. Then ui = (1/σi)Avi , for i = 1, 2, . . . , r, are the left singular vectors and by Theorem 1.5, A can bedecomposed into a sum of rank one matrices a

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T.$$

We first prove a simple lemma stating that two matrices A and B are identical if Av = Bv for all v. The lemma states that in the abstract, a matrix A can be viewed as a transformation that maps vector v onto Av

CHAPTER 6

RESULTS AND DISCUSSION

Since our project is movie recommendation system .one can develop a movie recommendation system by using either content based or collaborative filtering or combining both i.e hybrid model and popularity based

In our project we have developed a hybrid approach i.e. combination of both content and collaborative filtering .Both the approaches have advantages and dis-advantages .in content filtering based on the user ratings or user likes only such kind of movie will recommended to the user.

Advantages: it is easy to design and it takes less time to compute

Dis-advantages: the model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

In Collaborative filtering the recommendation is comparison of similar users.

Advantages: No need domain knowledge because the embeddings are automatically learned. The model can help users discover new interests. In isolation, the ML systemmay not know the user is interested in a given item, but the model might still recommend it because similar users are interested in that item.

Dis-advantages: The prediction of the model for a given (user, item) pair is the dot product of the corresponding embeddings. So, if an item is not seen during training, the system can't create an embedding for it and can't query the model with this item. This issue is often called the **cold-start problem.**

The hybrid approach will resolves all these limitations by combining both contentand collaborative filtering

The main disadvantage in hybrid approach is it require high memory

Deployment of Web-Application:

Here we tried to build the web application using stream lit for 4 types if recommendation system. As we are trying to deploy the application considering the whole data, we getting large size pickle file for it around 2GB. So, due to that we have not deployed our model on cloud. Due to shorter time for task completion we have not modified our model code for sampling data.
We will do this in further steps.

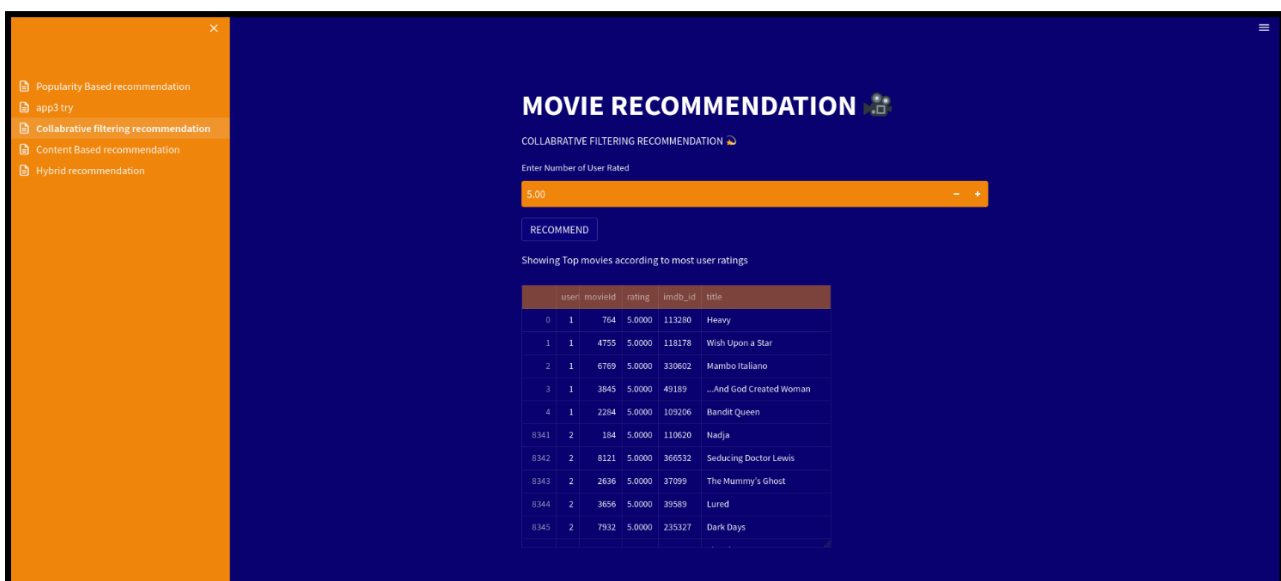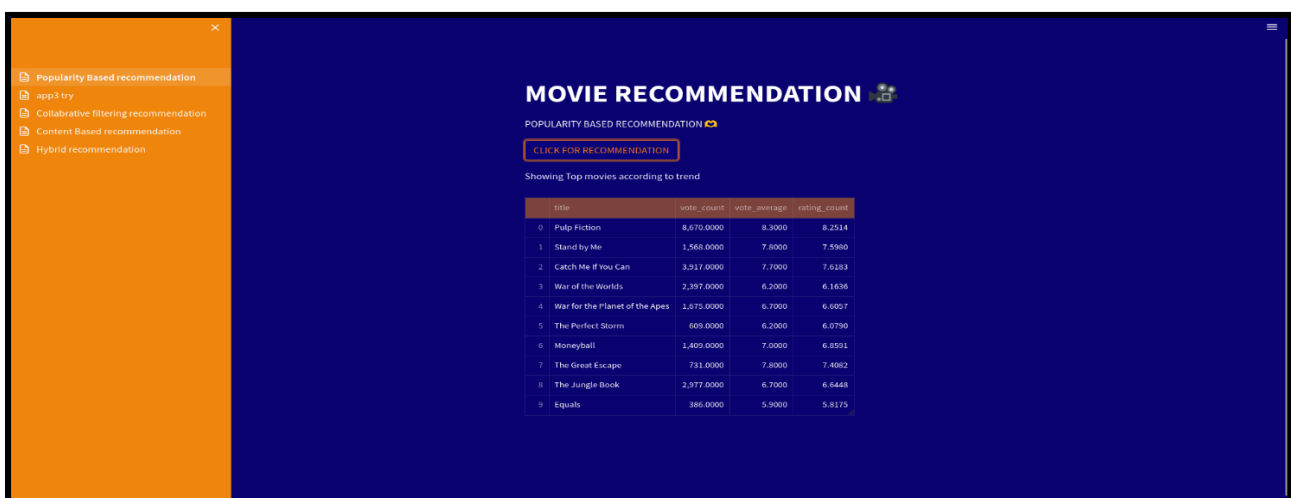**Below are the sample web pages which we have designed for all models.**

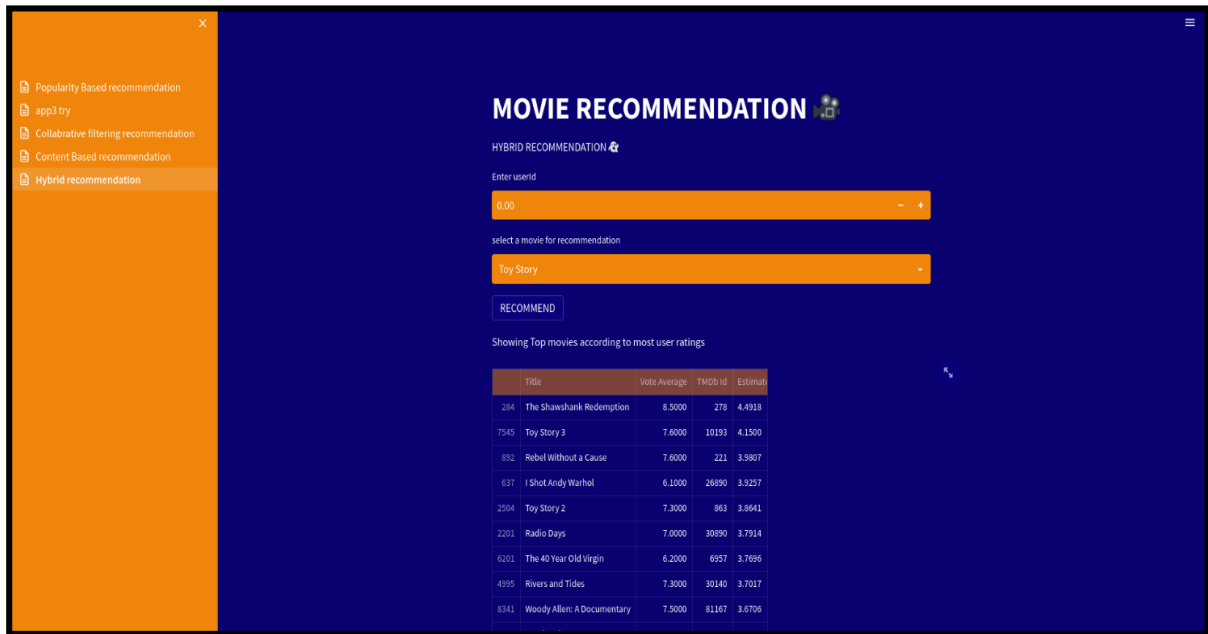**Fig:-6.2 user id window**

Enter the user id ranges between 1-5000

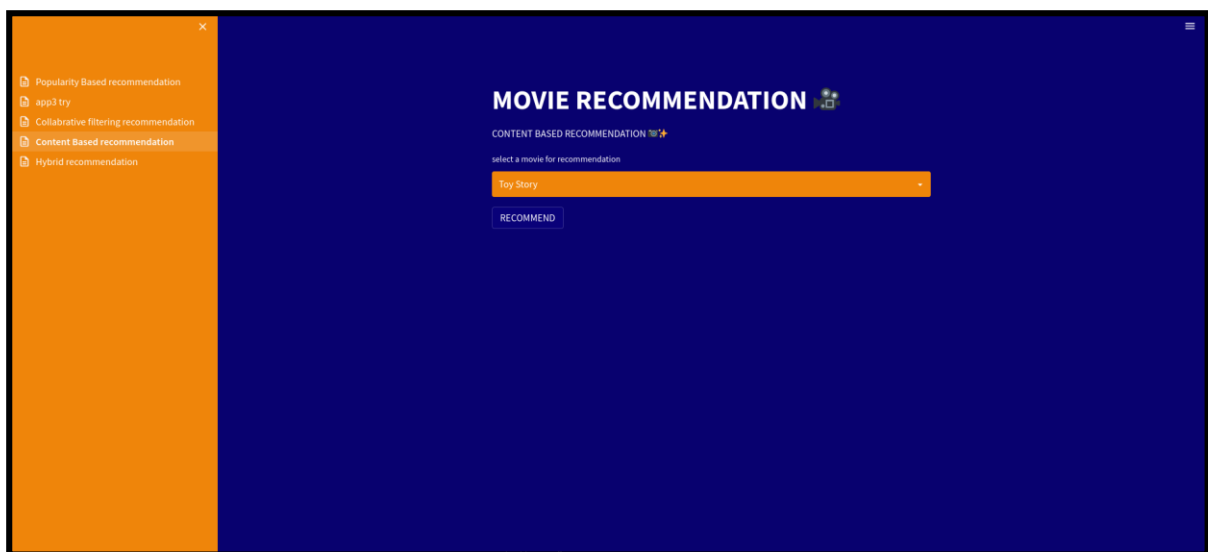**Fig: -6.3 Display of list of recommended movies**

Fig: -6.4. Display content based recommendation for movie name

CHAPTER 7
CONCLUSION AND FUTRURE SCOPE

7.1   Conclusion

In this project, to improve the accuracy, quality and scalability of movie recommendation system, a Hybrid approach by unifying content based filtering and collaborative filtering; using Singular Value Decomposition (SVD) as a classifier and Cosine Similarity is presented in the proposed methodology. Existing pureapproaches and proposed hybrid approach is implemented on three different Movie datasets and the results are compared among them. Comparative results depicts that the proposed approach shows an improvement in the accuracy, quality and scalability of the movie recommendation system than the pure approaches. Also, computing timeof the proposed approach is lesser than the other two pure approaches.

7.2   Future scope:

In the proposed approach, It has considered Genres of movies but, in future we can also consider age of user as according to the age movie preferences also changes, likefor example, during our childhood we like animated movies more as compared to other movies. There is a need to work on the memory requirements of the proposed approach in the future. The proposed approach has been implemented here on different movie datasets only. It can also be implemented on the Film Affinity and Netflix datasets and the performance can be computed in the future.

REFERENCES

- http://www.acm.org/cacm/MAR97/resnick.html

- http://www.ercim.org/publication/ws-proceedings/DelNoe02/CliffordLynchAbstract.pdf

- http://people.cs.vt.edu/~ramakris/papers/ppp.pdf

- http://www.cs.umn.edu/Research/GroupLens/papers/pdf/ec-99.pdf

- http://www.rashmisinha.com/talks/Recommenders-SIGIR.pdf

- https://youtu.be/v_mONWiFv0k

- https://towardsdatascience.com/recommendation-system-in-python-lightfm-61c85010ce17

- https://www.analyticsvidhya.com/blog/2022/01/a-comprehensive-guide-on-recommendation-engines-in-2022/

- https://en.wikipedia.org/wiki/Item-item_collaborative_filtering