Movie Recommendation System

A MINOR PROJECT REPORT

Submitted by

Monish Kumar[RA2011032010015]

Dhanush Kumar[RA2011032010021]

Under the guidance of

Dr. Balasaraswathi V R

(Assistant Professor Department of networking and communications)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M.Nagar, Kattankulathur, Chengalpattu District

April 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR603 203

BONAFIDE CERTIFICATE

Certified that this B. Tech – 18CSE392T project titled "Movie Recommendation system" is the Bonafide work of Dhanush Kumar (RA2011032010021), Monish Kumar (RA2011032010015) who carried out the project work under our supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Signature

r. Balasaraswathi V R
SUPERVISOR
ssistant Professor
epartment of Networking and
Communications

Dept. of Networking and Communications
Kattankulathur
603 203
SRM IST

Signature

DR. ANNAPURANI PANAIYAPPAN K.
HEAD OF THE DEPARTMENT
Professor
Department of Networking and
Communications

ABSTRACT

The recommendation system is one of the best subfields of information retrieval approaches where the large datasets are used to make meaningful predictions. These recommendation systems have been widely used with two approaches named content based andcollaborative filtering algorithm. The proposed system makes use of both the approaches to provide the combined algorithm which takes the user query and creates the results as movies that are related. By using the user profile, the hybrid recommendation system combines both content based and collaborative filtering algorithms that predict the users interested movies. The movie dataset is divided into training and testing data sets where the recommendation model is applied on the testing set to find the predictions. As part of the Content based filtering the item features are taken to predict the items related to user's query. The item features that areconsidered for the proposed system are the movie genres and ratings. In collaborative filtering process, the system uses the cosine similarity metric to find the distance between one user to other users. This approach overcomes the problem of cold start by using content based approach and handling data sparsity by selecting features. By combining both the recommendation algorithms, an hybrid approach is generated which provides more accurate results. This hybrid recommendation system finally recommends the set of movies that are relevant to the users interest.

Table of content

LIST OF FIGURES

ABBREVIATIONS

SQL	**Structed query language**
Pa	**Pandas**
St	**Streamlit**
WAN	**Wide Area Network**
DB	**Data Base**
LAN	**Local Area Network**
ML	**Machine Learning**

CHAPTER1

1. INTRODUCTION

In recent years, improving the effectiveness of the commercial web services especially to use the personalized recommendation technology to realize the electronic commerce personalized service has gradually become a hot topic can cause widespread interest. But at present domestic most of ecommerce recommendation is usually: recommended best-selling products. Recommend related product according to user's browsing history is recommended so to speak, the first two recommended due to fundamental not considering the personality traits of the different users, therefore recommend simply does not have the characteristics of individuation, the third recommend a personalized composition,but most of the site also only stay in only the users against a person's purchase history, just for each user set up a personal purchase records, no transverse to the comprehensive information, so there is no collaboration recommended value which also is unable to realizes real-time comprehensive recommended goods.According to the literature review, the modern optimization for the traditional recommendation systems could be summarized as the follows. To strengthen the user control. Most existing recommendation system according to preset automatically generates multiple recommended users' personal information and requirements, to some extent, limits the user participation and control. The result of the recommendation system should allow users to participate in parameter definition. Recommendation system 2 can be achieved by the relevance feedback mechanism to update the user's real-time demand , for example, by the user to explicitly to the evaluation of the recommended collect user feedback information.To strengthen the interactive interface design. System interface is one of the most important factors affecting the user satisfaction. Existing recommendation system most committed to the improvement of recommendation algorithm, paid little attention to interface problem. Using the multidimensional information visualization technology can make result of the recommended straightforward graphic interpretation can help users understand the causes of recommended. Recommended model support package.Recommendation model based on package, we put forward concept of incremental recommendationsystem, recommendation process is broken down into several successive steps, system recommended was generated according to the user requirements in each step, by the user decides to join the final plan of object,the user affect the choice of the system in the subsequent steps of recommendations

1.2. SYSTEM REQUIREMENTS SPECIFICATIONS

1.2.1 FUNCTIONAL REQUIREMENTS

Processor : Quad core with speed at least 3 GHz (i5 or i7 Processor)

RAM : 8GB or more

ROM : SSD more than 256 GB is preferred

Resolution : 1920 x 1080 or higher

Operating system : windows 7 or higher, macOS

Languages : python with libraries pickle, sikitlearn, streamlit, request, pandas must be installed

1.2.2 NON-FUNCTIONAL REQUIREMENTSUser Interfaces

The user interface for the software shall be compatible with any browser such as Internet Explorer, Mozilla, and Google Chrome by which users can access the system

Hardware Interfaces

Since the application must run over the internet, all the hardware required to be connected to the internet will be a hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross-Cable.

Software Interfaces

We have chosen Windows operating system for its best support and user-friendliness. To save the movie details, users preferences etc, we have chosen SQL database. To implement the project we will use Python libraries, Javascript, Html etc.

3. SYSTEM ARCHITECTURE AND DESIGN

3.1 Architecture of recommendation system:

A Content-Based Recommender works by the data that we take from the user, either explicitly (rating) or implicitly (clicking on a link). By the data we create a user profile, which is then used to suggest tothe user, as the user provides more input or take more actions on the recommendation, the engine becomes more accurate.

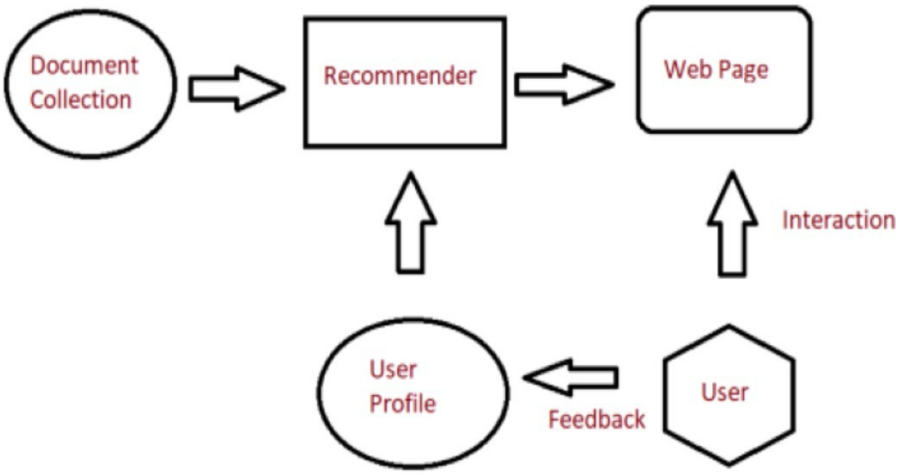


Fig:3.1 Architecture of recommendation system

User Profile:

In the User Profile, we create vectors that describe the user's preference. In the creation of a user profile, we use the utility matrix which describes the relationship between user and item. With this information, the bestestimate we can make regarding which item user likes, is some aggregation of the profiles of those items.

Item Profile:

In Content-Based Recommender, we must build a profile for each item, which will represent the

important characteristics of that item. For example, if we make a movie as an item then its actors, director, release year and genre are the most significant features of the movie. We can also add its rating from the IMDB(Internet Movie Database) in the Item Profile.

Utility Matrix:

Utility Matrix signifies the user's preference with certain items. In the data gathered from the user, wehave to find some relation between the items which are liked by the user and those which are disliked, for this purpose we use the utility matrix. In it we assign a particular value to each user-item pair, this value is known as the degree of preference. Then we draw a matrix of a user with the respective items to identify their preference relationship.

Users	Movie 1	Movie 2	Movie 3
User 1	3		1
User 2	2	4	

Fig: Utility Matrix of Movie Recommendation System

Fig 3.2 Utility Matrix

3.2 Recommendation system for movie:

The basic concept behind a movie recommendation system is quite simple. In particular, thereare two main elements in every recommender system: users and items. The system generates movie predictionsfor its users, while items are the movies themselves.

The primary goal of movie recommendation systems is to filter and predict only those movies that a corresponding user is most likely to want to watch. The ML algorithms for these recommendation systems usethe data about this user from the system's database.

This data is used to predict the future behavior of the user concerned based on the information from the past. Because data plays such an important role in ML projects, including the movie recommendation system, it should be handled by professionals. Contact our team of qualified data annotators at Label Your Data to ensureyour data is in good hands for ultimate success in AI!

Content-Based Filtering

A filtration strategy for movie recommendation systems, which uses the data provided about theitems (movies). This data plays a crucial role here and is extracted from only one user. An ML algorithm used for this strategy recommends motion pictures that are similar to the user's preferences in the past.

Therefore, the similarity in content-based filtering is generated by the data about the past film selections andlikes by only one user. How does it work? The recommendation system modellin the past preferences of the user concerned, and then it uses this information to try to find similar movies. This information is available in the database (e.g., lead actors, director, genre, etc.). After that, the system provides movie recommendations forthe user. That said, the core element in content-based filtering is only the data of only one user that is used to make predictions. —

Collaborative Filtering

As the name suggests, this filtering strategy is based on the combination of the relevant user's and other users' behaviors. The system compares and contrasts these behaviors for the most optimal results. It'sa collaboration of the multiple users' film preferences and behaviors. What's the mechanism behind this strategy? The core element in this movie recommendation system and the ML algorithm it's built on is the history of all users in the database. Basically, collaborative filtering is based on the interaction of all users in the system with the items (movies). Thus, every user impacts the final outcome of this ML-based recommendation system, while content-based filtering depends strictly on the data from one user for its modelling.

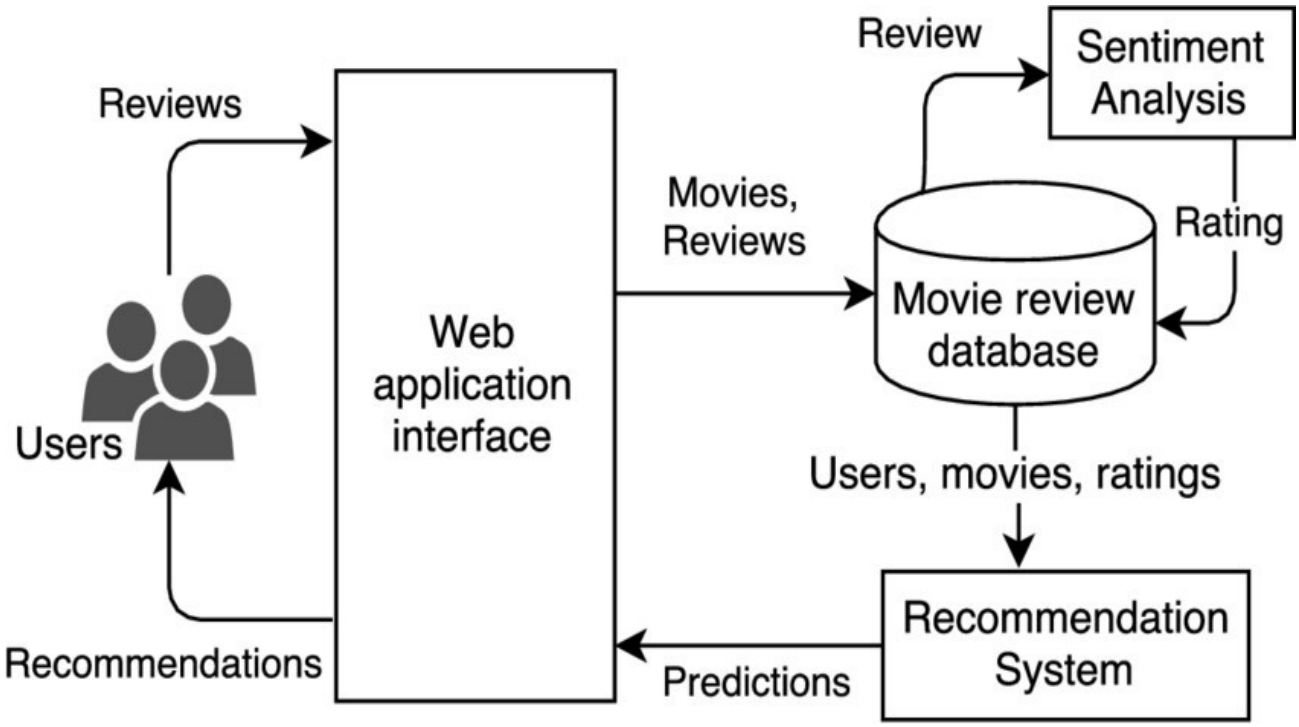


Fig 3.3 Movie recommendation system

4. METHODOLOGY

Movie recommendation systems use a set of different filtration strategies and algorithms to helpusers find the most relevant films. The most popular categories of the ML algorithms used for movie recommendations include

- **content-based filtering**
- **collaborative filtering systems.**

Content-Based Filtering

A filtration strategy for movie recommendation systems, which uses the data provided aboutthe items (movies). This data plays a crucial role here and is extracted from only one user. An ML algorithm used for this strategy recommends motion pictures that are similar to the user's preferences in the past.

Therefore, the similarity in content-based filtering is generated by the data about the past film selections and likes by only one user. How does it work? The recommendation system analyzes the past preferences of the user concerned, and then it uses this information to try to find similar movies. This information is available in the database (e.g., lead actors, director, genre, etc.). After that, the system provides movie recommendations for the user. That said, the core element in content-based filtering is only the data of only one user that is used to make predictions.

Collaborative Filtering

As the name suggests, this filtering strategy is based on the combination of the relevant user's andother users' behaviors. The system compares and contrasts these behaviors for the most optimal results. It's a collaboration of the multiple users' film preferences and behaviors. What's the mechanism behind this strategy? The core element in this movie recommendation system and the ML algorithm it's built on is the history of all users in the database. Basically, collaborative filtering is based on the interaction of all users in the system with the items (movies). Thus, every user impacts the final outcome of this ML-based recommendation system, while content-based filtering depends strictly on the data from one user for its modeling.

Collaborative filtering algorithms are divided into two categories:

User-based collaborative filtering.

The idea is to look for similar patterns in movie preferences in the target user and other users in the database.

Item-based collaborative filtering.

The basic concept here is to look for similar items (movies) that target users rate or interact with. The modern approach to the movie recommendation systems implies a mix of both strategies for the mostgradual and explicit results.

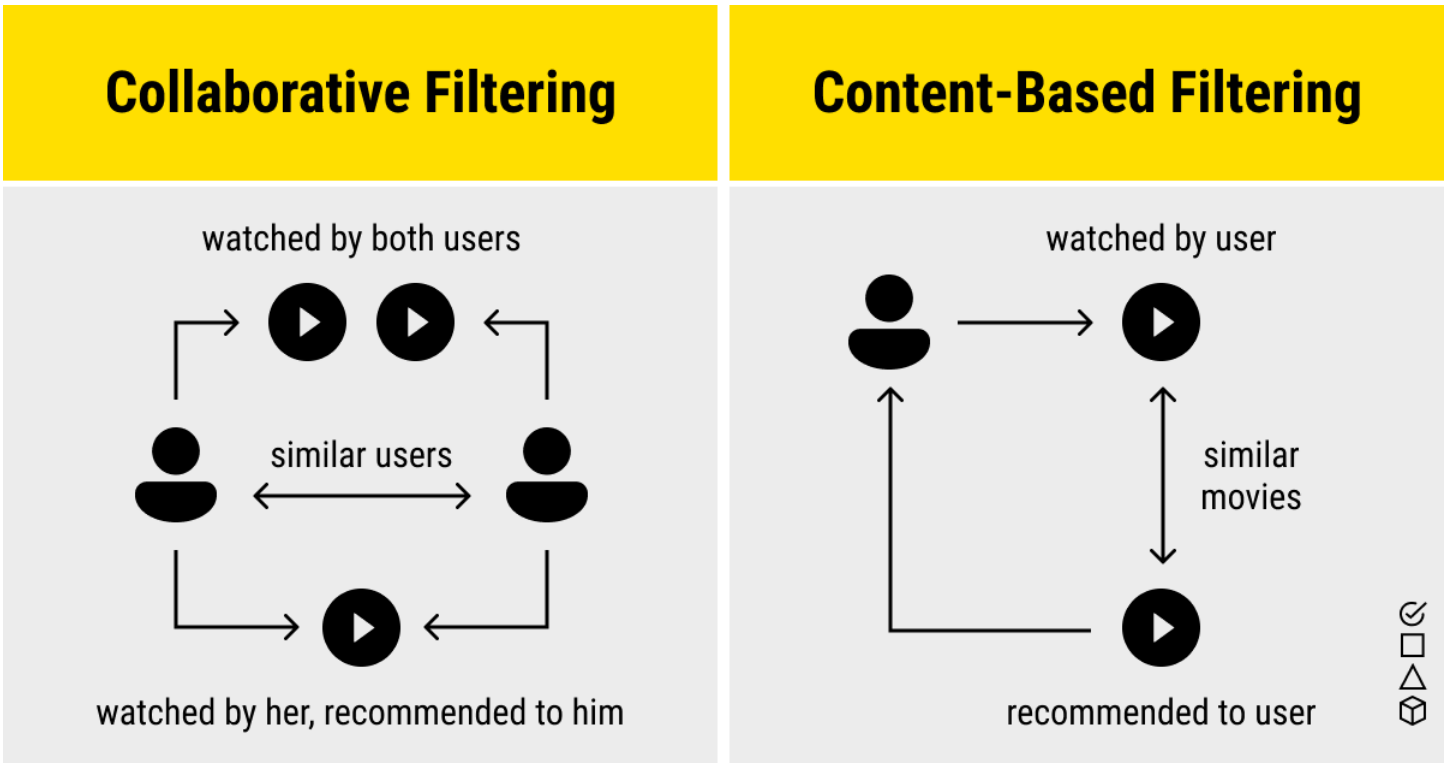
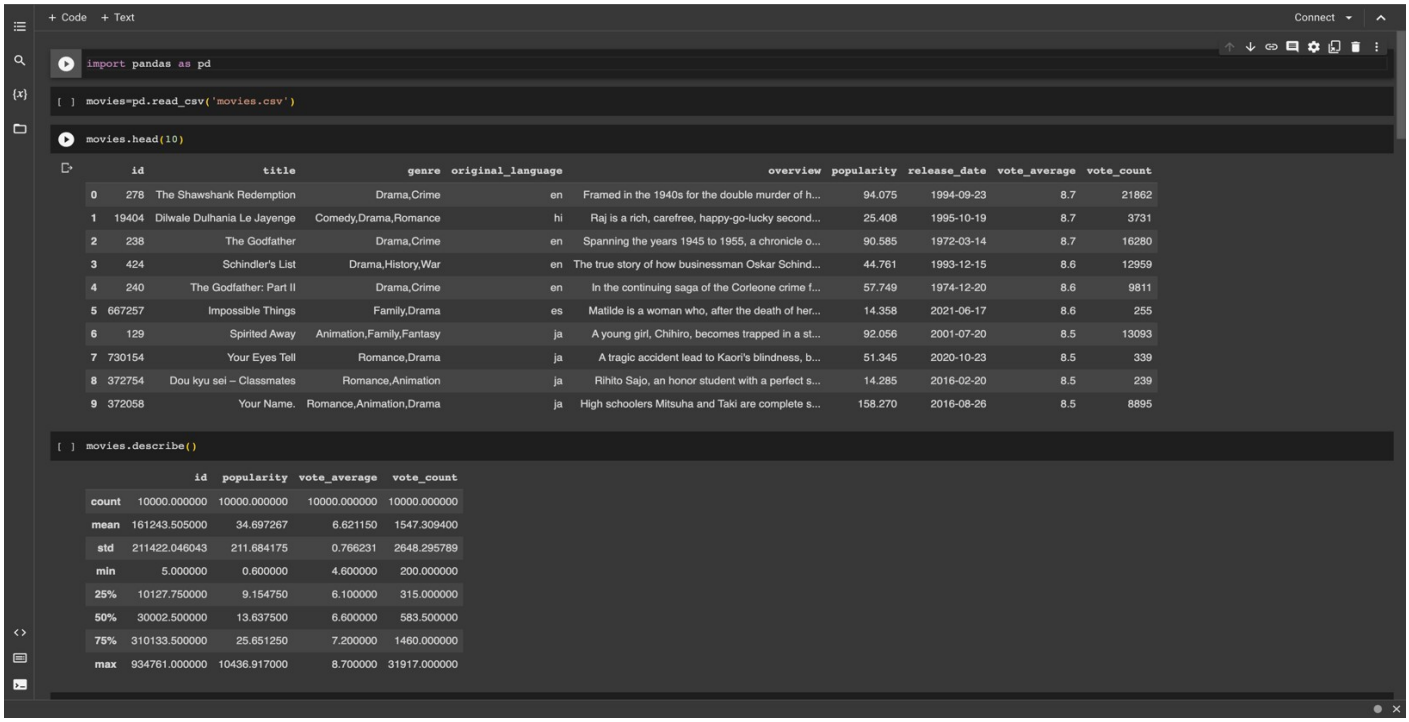


Fig 4.1 Collaborative vs Content-Based Filtering

5. CODING AND TESTING

Code for importing the dataset and recommend movie based on the dataset



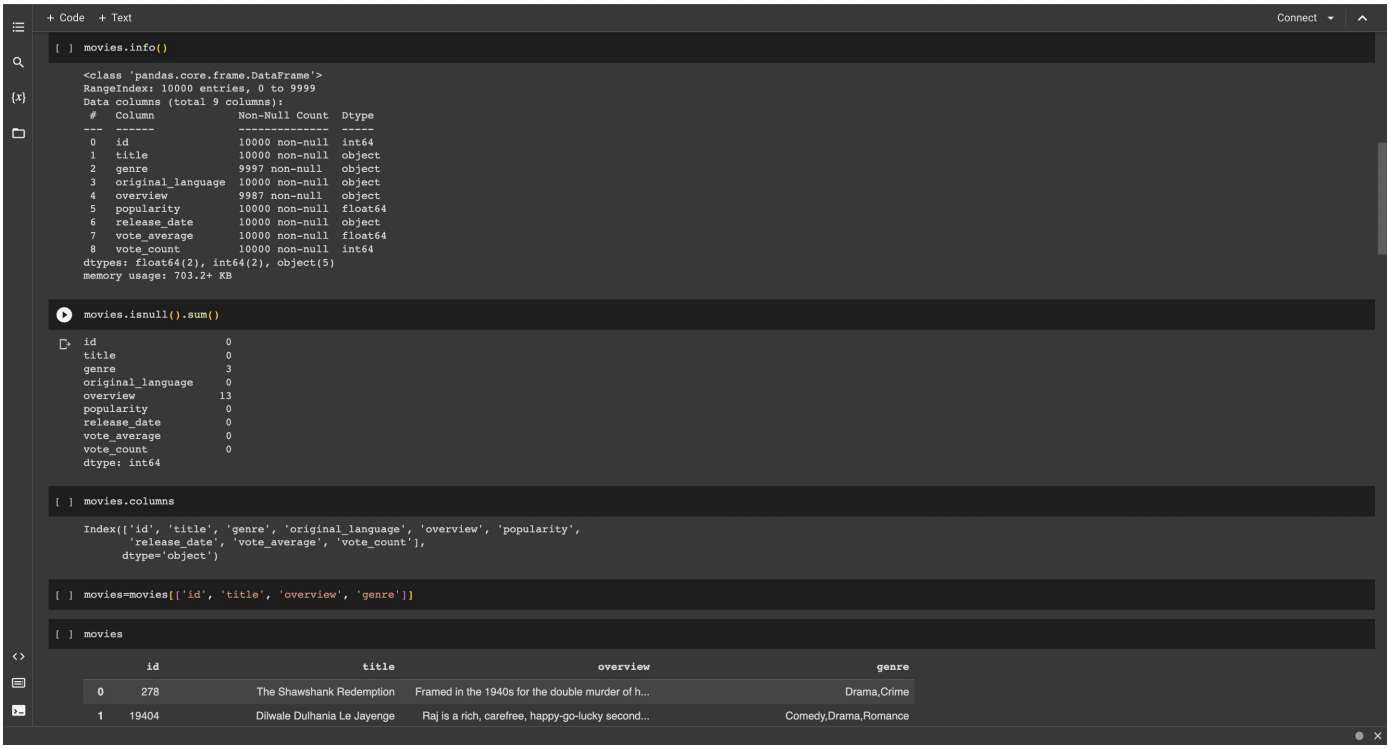
```
+ Code + Text                                                    Connect
import pandas as pd
[ ] movies=pd.read_csv('movies.csv')
movies.head(10)
```

	id	title	genre	original_language	overview	popularity	release_date	vote_average	vote_count
0	278	The Shawshank Redemption	Drama,Crime	en	Framed in the 1940s for the double murder of h...	94.075	1994-09-23	8.7	21862
1	19404	Dilwale Dulhania Le Jayenge	Comedy,Drama,Romance	hi	Raj is a rich, carefree, happy-go-lucky second...	25.408	1995-10-19	8.7	3731
2	238	The Godfather	Drama,Crime	en	Spanning the years 1945 to 1955, a chronicle o...	90.585	1972-03-14	8.7	16280
3	424	Schindler's List	Drama,History,War	en	The true story of how businessman Oskar Schind...	44.761	1993-12-15	8.6	12959
4	240	The Godfather: Part II	Drama,Crime	en	In the continuing saga of the Corleone crime f...	57.749	1974-12-20	8.6	9811
5	667257	Impossible Things	Family,Drama	es	Matilde is a woman who, after the death of her...	14.358	2021-06-17	8.6	255
6	129	Spirited Away	Animation,Family,Fantasy	ja	A young girl, Chihiro, becomes trapped in a st...	92.056	2001-07-20	8.5	13093
7	730154	Your Eyes Tell	Romance,Drama	ja	A tragic accident lead to Kaori's blindness, b...	51.345	2020-10-23	8.5	339
8	372754	Dou kyu sei – Classmates	Romance,Animation	ja	Rihito Sajo, an honor student with a perfect s...	14.285	2016-02-20	8.5	239
9	372058	Your Name.	Romance,Animation,Drama	ja	High schoolers Mitsuha and Taki are complete s...	158.270	2016-08-26	8.5	8895

```
[ ] movies.describe()
```

	id	popularity	vote_average	vote_count
count	10000.000000	10000.000000	10000.000000	10000.000000
mean	161243.505000	34.697267	6.621150	1547.309400
std	211422.046043	211.684175	0.766231	2648.295789
min	5.000000	0.600000	4.600000	200.000000
25%	10127.750000	9.154750	6.100000	315.000000
50%	30002.500000	13.637500	6.600000	583.500000
75%	310133.500000	25.651250	7.200000	1460.000000
max	934761.000000	10436.917000	8.700000	31917.000000

Fig 5.1 importing the dataset



```
+ Code + Text                                                    Connect
[ ] movies.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 10000 non-null  int64
 1   title              10000 non-null  object
 2   genre              9997 non-null   object
 3   original_language  10000 non-null  object
 4   overview           9987 non-null   object
 5   popularity         10000 non-null  float64
 6   release_date       10000 non-null  object
 7   vote_average       10000 non-null  float64
 8   vote_count         10000 non-null  int64
dtypes: float64(2), int64(2), object(5)
memory usage: 703.2+ KB
movies.isnull().sum()
id                   0
title                0
genre                3
original_language    0
overview            13
popularity           0
release_date         0
vote_average         0
vote_count           0
dtype: int64
[ ] movies.columns
Index(['id', 'title', 'genre', 'original_language', 'overview', 'popularity',
       'release_date', 'vote_average', 'vote_count'],
      dtype='object')
[ ] movies=movies[['id', 'title', 'overview', 'genre']]
[ ] movies
```

	id	title	overview	genre
0	278	The Shawshank Redemption	Framed in the 1940s for the double murder of h...	Drama,Crime
1	19404	Dilwale Dulhania Le Jayenge	Raj is a rich, carefree, happy-go-lucky second...	Comedy,Drama,Romance

Fig 5.2 trim the required columns

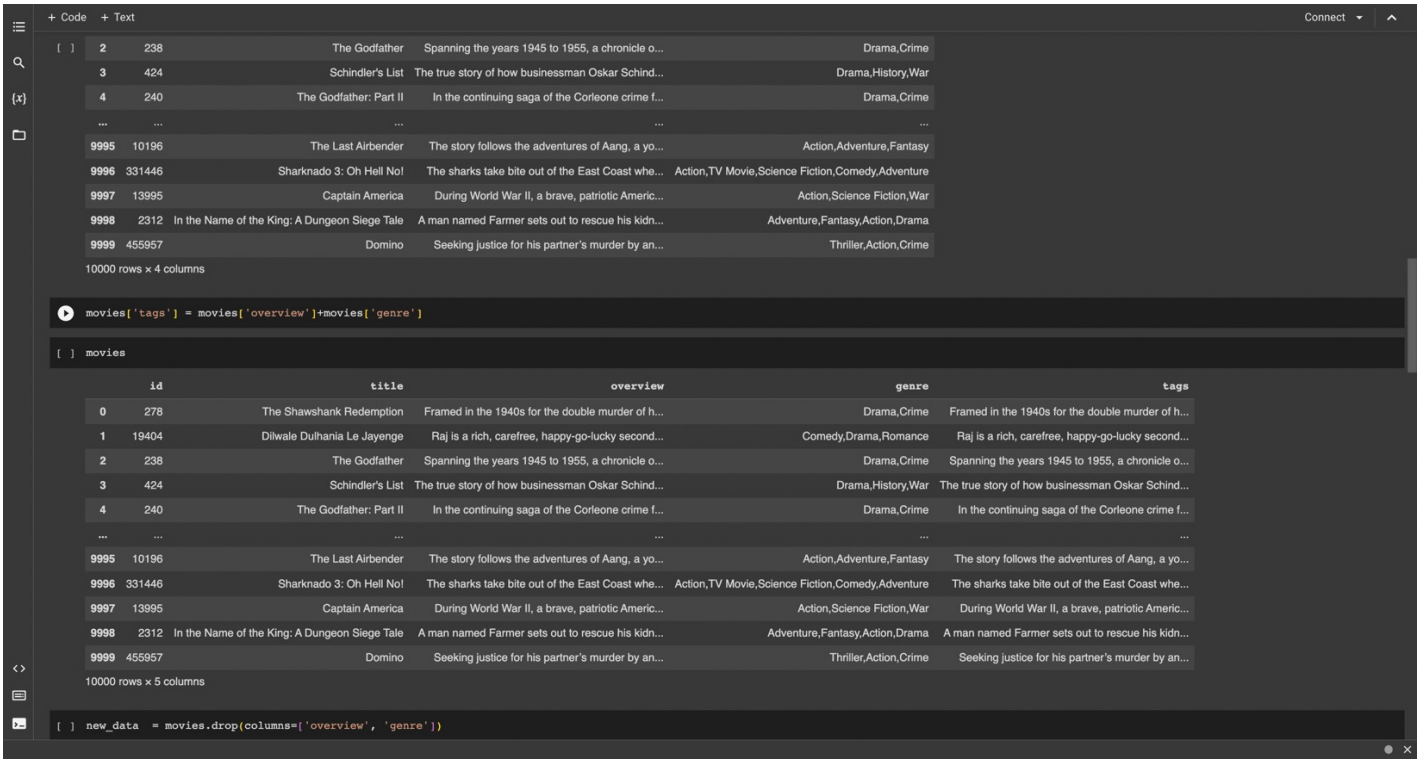


Fig 5.3 combining the columns

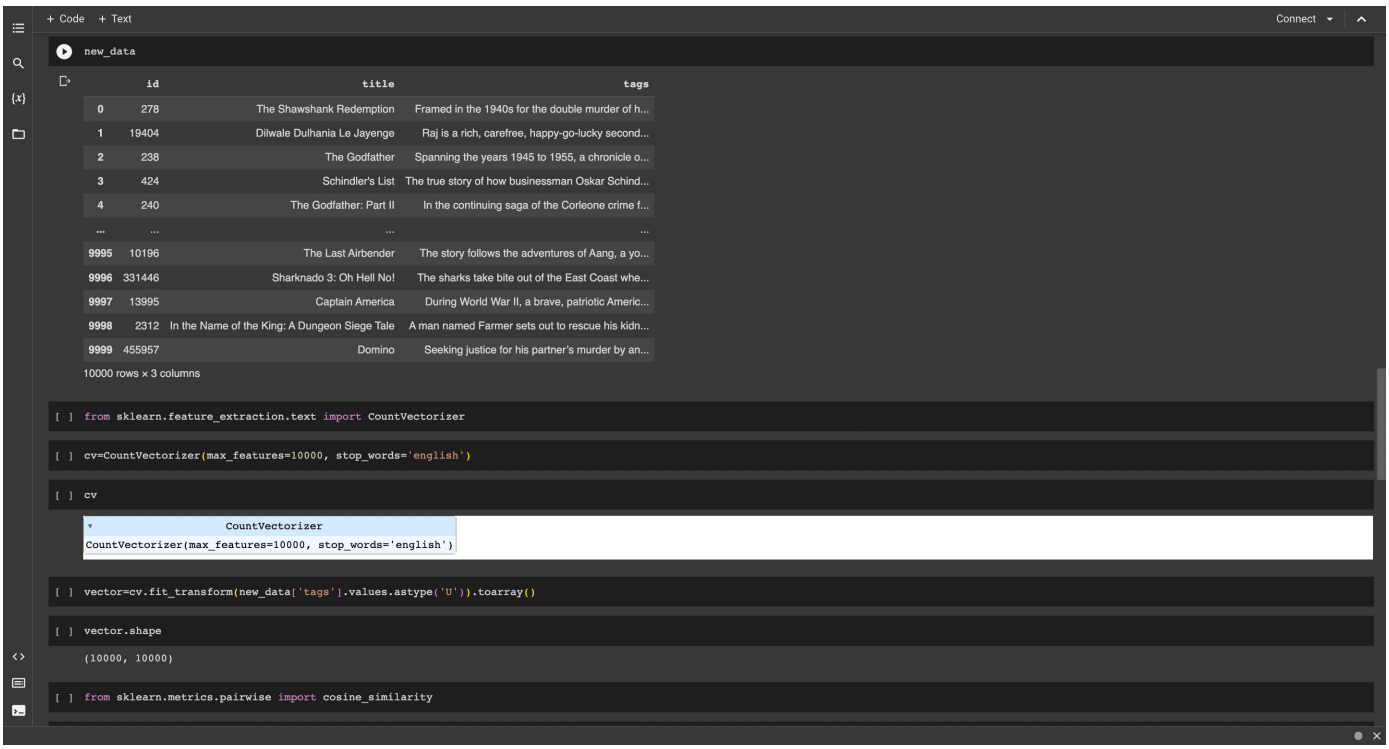
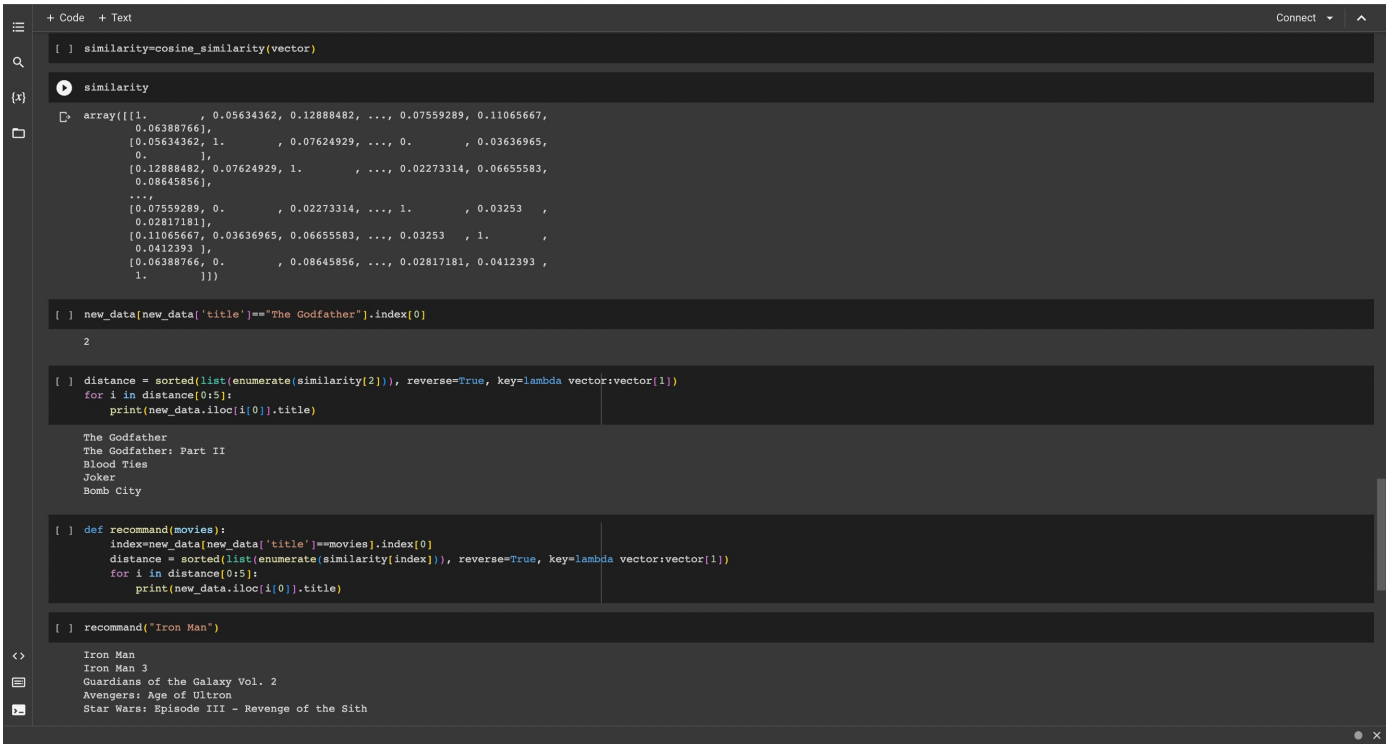


Fig 5.4 importing vector



```
similarity=cosine_similarity(vector)
```

```
similarity
```

```
array([[1.        , 0.05634362, 0.12888482, ..., 0.07559289, 0.11065667,
        0.06388766],
       [0.05634362, 1.        , 0.07624929, ..., 0.        , 0.03636965,
        0.        ],
       [0.12888482, 0.07624929, 1.        , ..., 0.02273314, 0.06655583,
        0.08645856],
       ...,
       [0.07559289, 0.        , 0.02273314, ..., 1.        , 0.03253   ,
        0.02817181],
       [0.11065667, 0.03636965, 0.06655583, ..., 0.03253   , 1.        ,
        0.0412393 ],
       [0.06388766, 0.        , 0.08645856, ..., 0.02817181, 0.0412393 ,
        1.        ]])
```

```
new_data[new_data['title']=="The Godfather"].index[0]
```

```
2
```

```
distance = sorted(list(enumerate(similarity[2])), reverse=True, key=lambda vector:vector[1])
for i in distance[0:5]:
    print(new_data.iloc[i[0]].title)
```

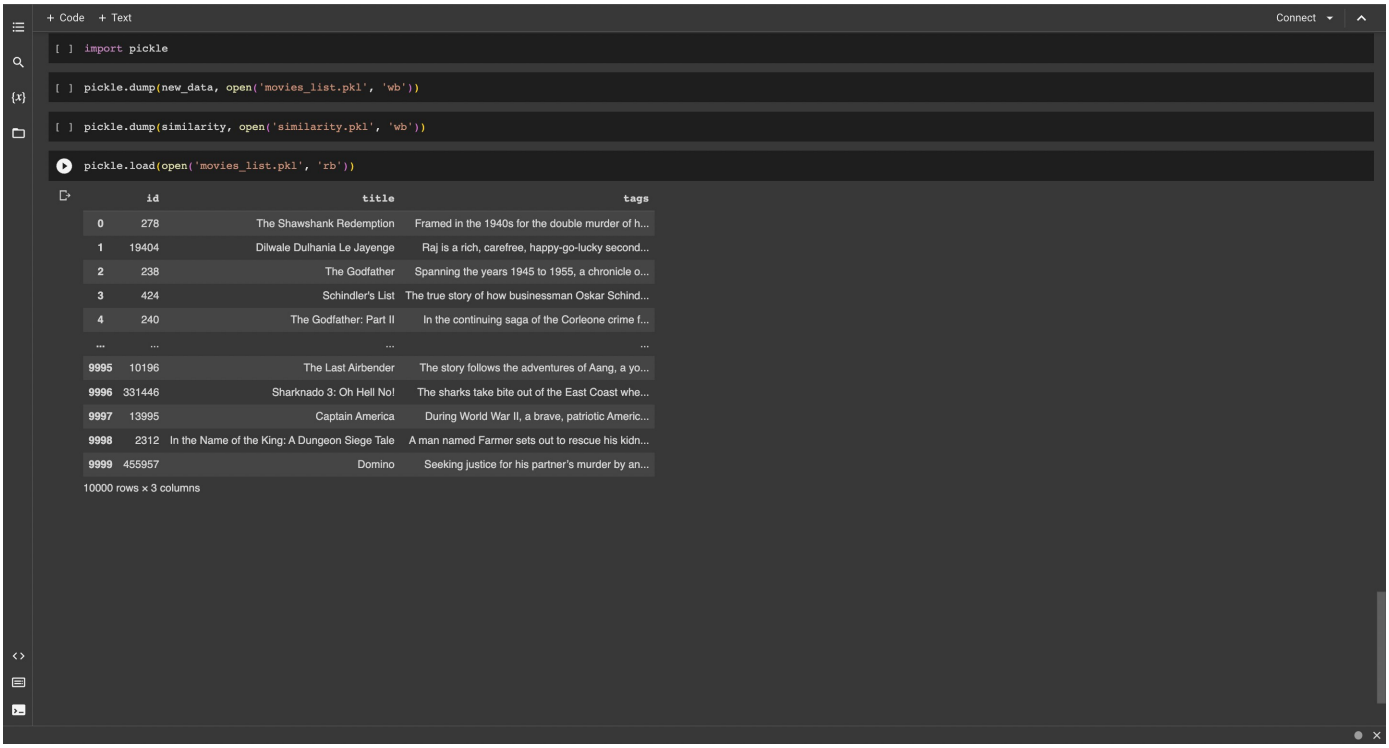
```
The Godfather
The Godfather: Part II
Blood Ties
Joker
Bomb City
```

```
def recommand(movies):
    index=new_data[new_data['title']==movies].index[0]
    distance = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda vector:vector[1])
    for i in distance[0:5]:
        print(new_data.iloc[i[0]].title)
```

```
recommand("Iron Man")
```

```
Iron Man
Iron Man 3
Guardians of the Galaxy Vol. 2
Avengers: Age of Ultron
Star Wars: Episode III - Revenge of the Sith
```

Fig 5.5 function for recommendation



```
import pickle
```

```
pickle.dump(new_data, open('movies_list.pkl', 'wb'))
```

```
pickle.dump(similarity, open('similarity.pkl', 'wb'))
```

```
pickle.load(open('movies_list.pkl', 'rb'))
```

	id	title	tags
0	278	The Shawshank Redemption	Framed in the 1940s for the double murder of h...
1	19404	Dilwale Dulhania Le Jayenge	Raj is a rich, carefree, happy-go-lucky second...
2	238	The Godfather	Spanning the years 1945 to 1955, a chronicle o...
3	424	Schindler's List	The true story of how businessman Oskar Schind...
4	240	The Godfather: Part II	In the continuing saga of the Corleone crime f...
...	...	...	...
9995	10196	The Last Airbender	The story follows the adventures of Aang, a yo...
9996	331446	Sharknado 3: Oh Hell No!	The sharks take bite out of the East Coast whe...
9997	13995	Captain America	During World War II, a brave, patriotic Americ...
9998	2312	In the Name of the King: A Dungeon Siege Tale	A man named Farmer sets out to rescue his kidn...
9999	455957	Domino	Seeking justice for his partner's murder by an...

10000 rows × 3 columns

Fig 5.6 importing pickle library

Code for creating the website and implement the recommendation function

```
import streamlit as st
import pickle
import requests

def fetch_poster(movie_id):
    url ="https://api.themoviedb.org/3/movie/{}?api_key=c7ec19ffdd3279641fb606d19ceb9bb1&language=en-US".format(movie_id)
    data=requests.get(url)
    data=data.json()
    poster_path = data['poster_path']
    full_path = "https://image.tmdb.org/t/p/w500/"+poster_path
    return full_path

movies = pickle.load(open("movies_list.pkl", 'rb'))
similarity = pickle.load(open("similarity.pkl", 'rb'))
movies_list=movies['title'].values

st.header("Movie Recommender System")



selectvalue=st.selectbox("Select movie from dropdown", movies_list)

def recommend(movie):
   index=movies[movies['title']==movie].index[0]
   distance = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda vector:vector[1])
   recommend_movie=[]
   recommend_poster=[]
   for i in distance[1:6]:
      movies_id=movies.iloc[i[0]].id
      recommend_movie.append(movies.iloc[i[0]].title)
      recommend_poster.append(fetch_poster(movies_id))
   return recommend_movie, recommend_poster
```

```
if st.button("Show Recommend"):
    movie_name, movie_poster = recommend(selectvalue)
    col1,col2,col3,col4,col5=st.columns(5)
    with col1:
        st.text(movie_name[0])
        st.image(movie_poster[0])
    with col2:
        st.text(movie_name[1])
        st.image(movie_poster[1])
    with col3:
        st.text(movie_name[2])
        st.image(movie_poster[2])
    with col4:
        st.text(movie_name[3])
        st.image(movie_poster[3])
    with col5:
        st.text(movie_name[4])
        st.image(movie_poster[4])
```

6. RESULTS AND DISCUSSIONS

6.1 Result:

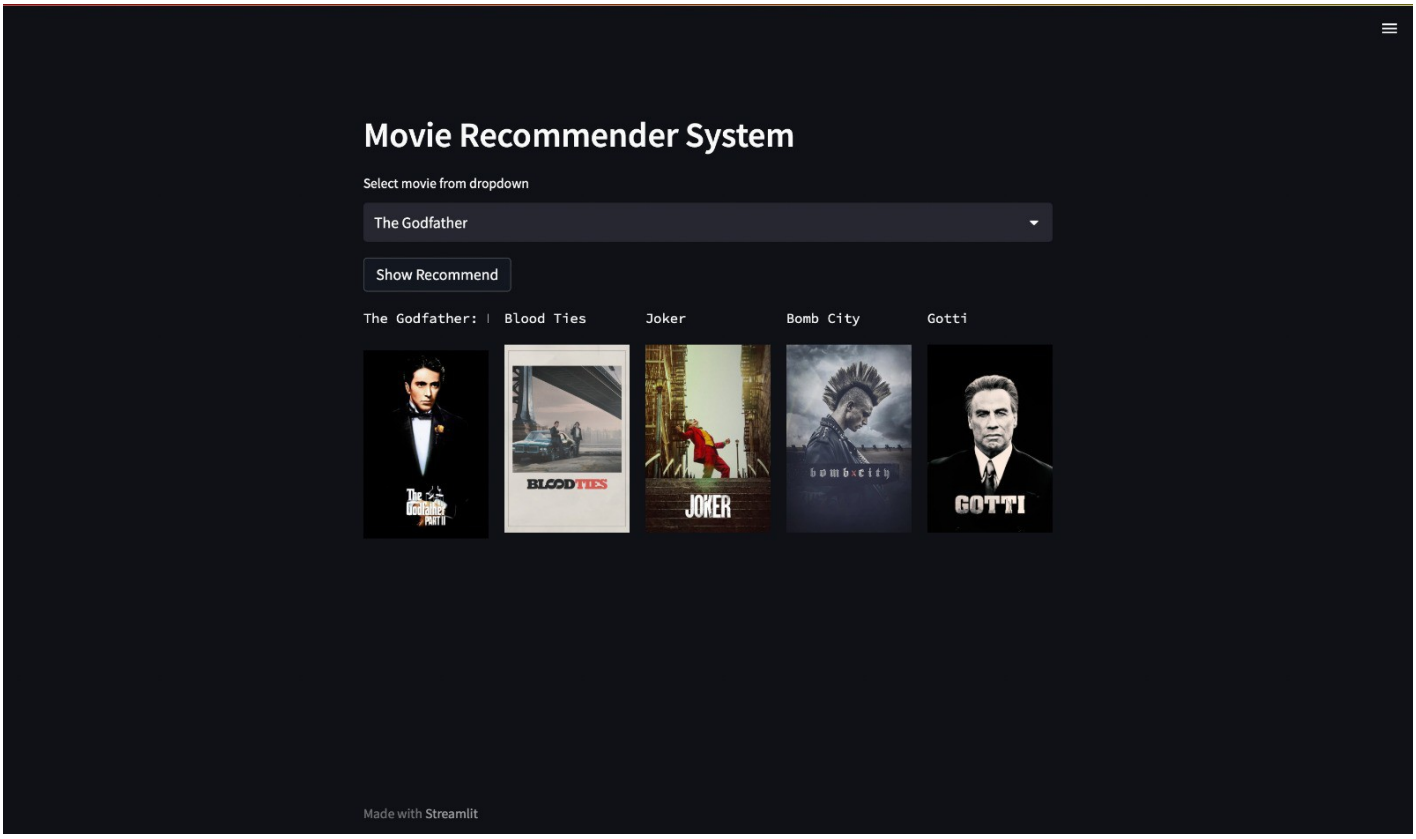


Fig 6.1 Output of the project

6.2 DISCUSSIONS

After the data processing, the keywords are extracted. Keyword extraction is one of the important process while working with the text. It is the task which the terms are automatically identified to describe the best subject in the document. Key phrases or key segments are the terms that tells about the relevant information contained in the document. Users reviews are collected as data set. The data set is pre-processed to remove the unwanted texts and missing values. The keywords are extracted from the pre-processed dataset.

The proposed system uses the Python programming for the implementation purposes. The recommendation is made where the recommendation engine provides the set of movies that are relevent to the

users profile. The overall dataset is divided as the training and test datasets. The training sets are used to apply the recommendation engine to predict the results and the results are then be checked with the model applied on the test dataset. The results obtained from the test results are then compared with the results derived from the train sets.

7. CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION:

Movie recommendation and selection is a fundamental issue in service oriented computing.

Existing works use either content based or collaborative filtering approaches to recommend movies to the users. Such approaches possess many limitations such as poor recommendation performance and heavily relying on the input from users. This approach exploits a three-way aspect model that systematically combines classic collaborative filtering and content-based recommendation. The proposed hybrid approach simultaneously considers the similarities of user ratings and semantic content of movies data. The experimentalresults show that our approach outperforms the conventional collaborative and content-based methods in termsof recommendation performance.

7.2 FUTURE ENHANCEMENT:

Future research will be done in two areas. First, includes exploring more refined/personalized recommendation by considering the specific contexts. This approach apply to other areas such as service clustering. Second, with respect to users, mining their implicit interests from usage records or reviews may be acomplement to the explicit interests (ratings). By this means, recommendations can be generated even if there are only few ratings. This will solve the sparsity problem to some extent.

REFERENCES:

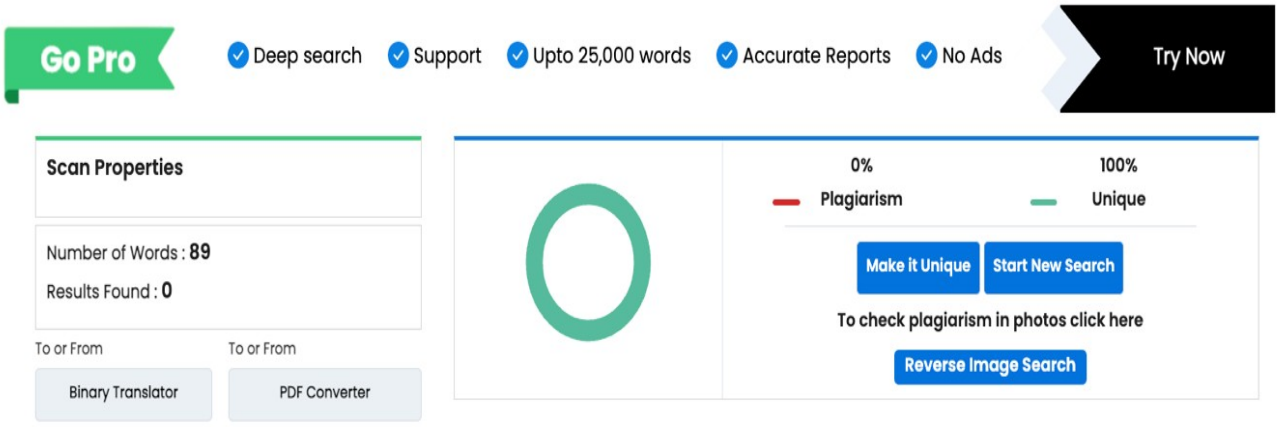
https://training.certstaff.com/News/108/System-Requirements-for-Web-Development#:~:text=Although%20most%20Web%20development%20tasks,processor%20might%20serve%2 0you%20well.

https://www.studocu.com/en-us/document/kipp-dc-college-preparatory/information-communication-and- technology/srs-bscit/47374918

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9269752/#:~:text=In%20movie%20recommender%20systems%2C%20the%20recommendations%20are%20made%20based%20on,%2C%20and%20ethnicity%20%5B14% 5D.

https://labelyourdata.com/articles/movie-recommendation-with-machine-learning#:~:text=Movie%20recommendation%20systems%20use%20a,filtering%20and%20collaborative%20fi ltering%20systems.

Results



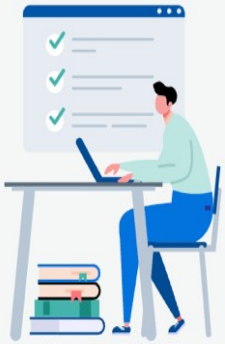

```
import streamlit as st import pickle
import requests


def fetch_poster(movie_id):
url ="https://api.themoviedb.org/3/movie/{}?
api_key=c7ec19ffdd3279641fb606d19ceb9bb1&language=en-
US".format(movie_id)
data=requests.get(url) data=data.json()
poster_path = data['poster_path']
full_path = "https://image.tmdb.org/t/p/w500/"+poster_path return
full_path

movies = pickle.load(open("movies_list.pkl", 'rb')) similarity =
pickle.load(open("similarity.pkl", 'rb')) movies_list=movies['title'].values

st.header("Movie Recommender System")

selectvalue=st.selectbox("Select movie from dropdown", movies_list) def
recommend(movie):
index=movies[movies['title']==movie].index[0]
```



Congratulations !

No Plagiarism Found