ONLINE MOVIE RECOMMENDATION SYSTEM

Chapter 1

INTRODUCTION

Movie Recommendation systems are information filtering tools that aspire to predict the rating for users and items, predominantly from big data to recommend their likes. Movie recommendation systems provide a mechanism to assist users in classifying users with similar interests. This makes movie recommender systems essentially a central part of websites and e-commerce applications. The primary objective of movie recommendation systems is to suggest a recommender system through data clustering and computational intelligence. We need a system to consolidate all these data into a representable, user friendly interface where the user can be recommended the best movie of his choice by quick filtering of the information on the web.

Chapter 2

Different types of recommendation engines

The most common types of recommendation systems are **content-based** and **collaborative filtering** recommender systems. In collaborative filtering, the behavior of a group of users is used to make recommendations to other users. The recommendation is based on the preference of other users. A simple example would be recommending a movie to a user based on the fact that their friend liked the movie. There are two types of collaborative models **Memory-based** methods and **Model-based** methods. The advantage of memory-based techniques is that they are simple to implement and the resulting recommendations are often easy to explain. They are divided into two:

- **User-based collaborative filtering**: In this model, products are recommended to a user based on the fact that the products have been liked by users similar to the user. For example, if Derrick and Dennis like the same movies and a new movie come out that Derick like, then we can recommend that movie to Dennis because Derrick and Dennis seem to like the same movies.
- **Item-based collaborative filtering**: These systems identify similar items based on users' previous ratings. For example, if users A, B, and C gave a 5-star rating to books X and Y then when a user D buys book Y they also get a recommendation to purchase book X because the system identifies book X and Y as similar based on the ratings of users A, B, and C.

Model-based methods are based on Matrix Factorization and are better at dealing with sparsity. They are developed using data mining, machine learning algorithms to predict users' rating of unrated items. In this approach techniques such as dimensionality reduction are used to improve accuracy. Examples of such model-based methods include Decision trees, Rule-based Model, Bayesian Model, and latent factor models.

Chapter 3

Datasets to use for building recommender systems

Dataset

- I choose the TMDb movie data set for data analysis. This data set contains information about 10,000 movies collected from The Movie Database (TMDb), including user ratings and revenue.I would like to find other intresting patterns in the dataset.

  Contain:

  - Total Rows = 10866
  - Total Columns = 21
  - After Seeing the dataset we can say that some columns is contain null values

Conclusions

- Drama is the most popular genre, following by action, comedy and thriller.
- Drame, Comedy, Thriller and Action are four most-made genres.
- Maximum Number Of Movies Release In year 2014.
- 'Avatar', 'Star Wars' and 'Titanic' are the most profitable movies.
- Short or Long duration movies are more popular than long duration movies.
- Average runtime of the movies are decreasing year by year.
- May,june,november and december are most popular month for releasing movies, if you want to earn more profit.

Technology used and Its Characteristics

Software Requirements:

- Backend Language: PYTHON.
- Editor: Pycharm, Jupyter notebook.

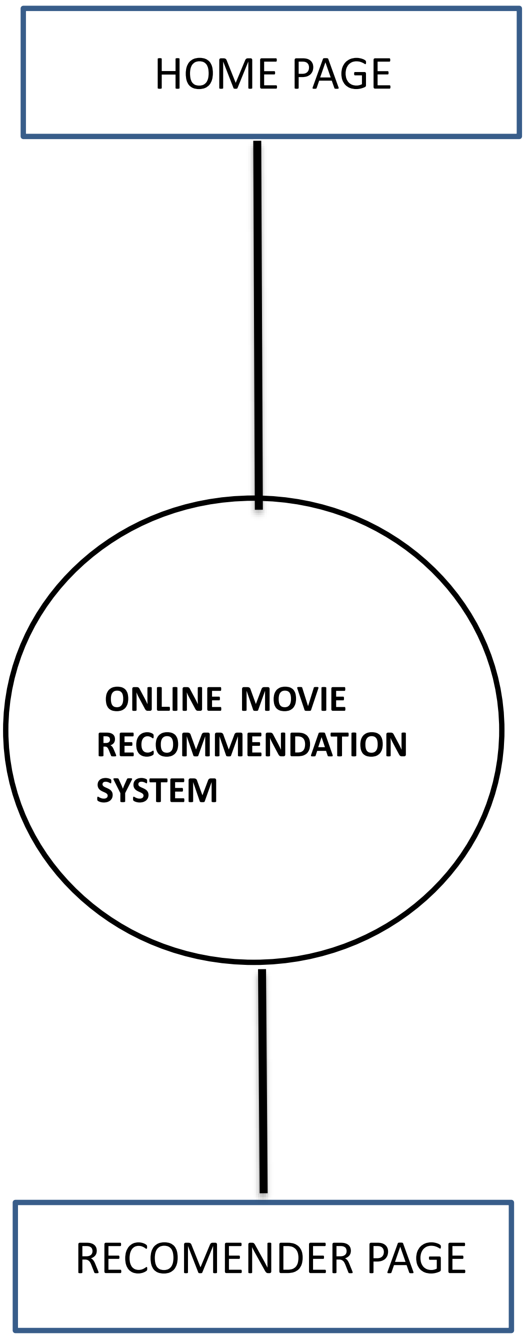
Hardware Requirements:

- Operating system: Windows 7 & above
- Ram: 2Gb and more
- Storage: min 50GB or more

Visual Studio Code

Chapter 5

PROPOSED SYSTEM



Chapter 6

Walkthrough of building a recommender system

We are going to use the movie lens to build a simple item similarity-based recommender system. The first thing we need to do is to import pandas and numpy.

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

Next, we load in the data set using pandas `read_csv()` utility. The dataset is tab separated so we pass in `\t` to the `sep` parameter. We then pass in the column names using the `names` parameter

- **Observation From The Dataset**
  - The columns 'budget', 'revenue', 'budget_adj', 'revenue_adj' has not given.But for this dataset i will assume the currency is in US dollor.
  - The dataset contain lots of movies where the budget or revenue have a value of '0'.

In [5]:

- **Data Cleaning (Removing The Unused Information From The Dataset)**

Information That We Need To Delete Or Modify

1. We need to remove duplicate rows from the dataset
2. Changing format of release date into datetime format

3. Remove the unused colums that are not needes in the analysis process.
4. Remove the movies which are having zero value of budget and revenue

- **Remove the unused colums that are not needes in the analysis process**

5. We can see that 21 columns in the dataset, We can drop the the colums which are not usable in the data analysis process. columns like: imdb_id,overview etc.
   The columns like imdb_id, homepage,tagline, overview, budget_adj and revenue_adj are not required for my analysis and I will drop these columns.

- **Drop theses rows which contain incorrect or inappropriate values.**

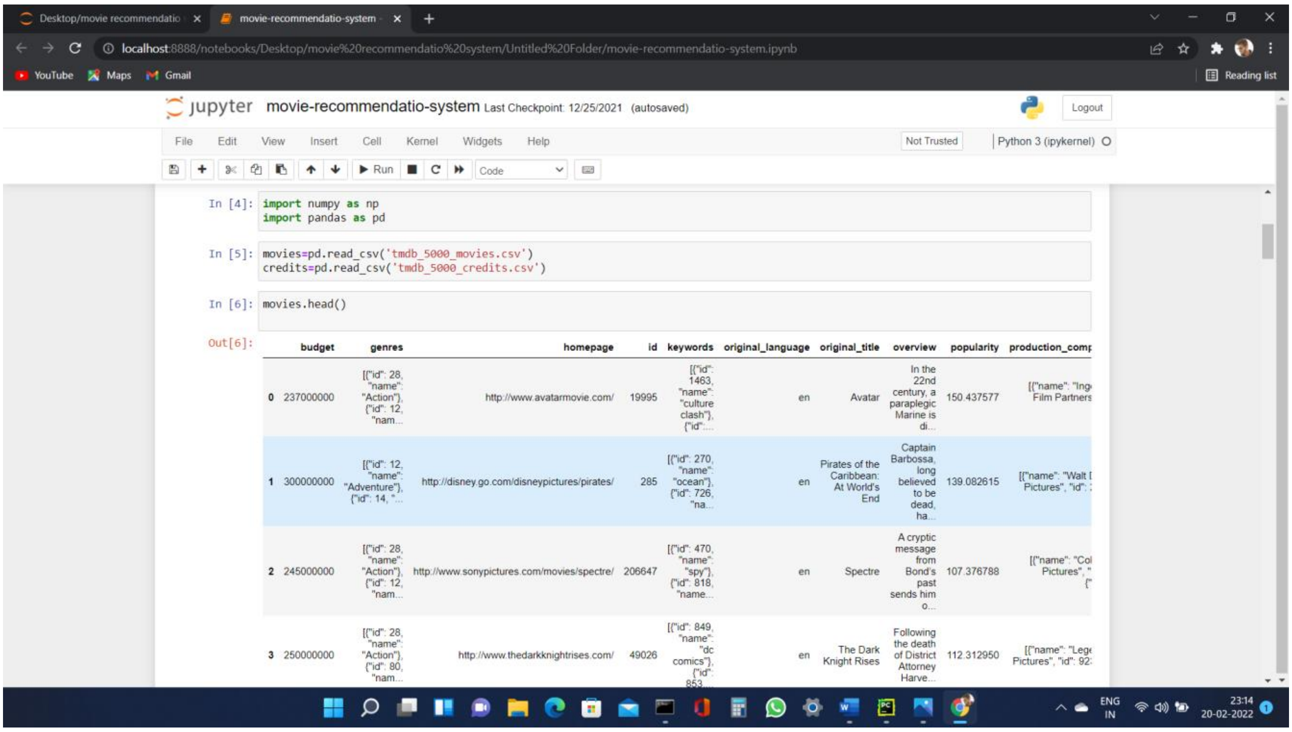
6. As you can see in this database of movies there are lots of movies where the budget or revenue have a value of '0' which means that the values of those variables of those movies has not been recorded. Calculating the profits of these movies would lead to inappropriate results. I think this may be due to varying factors like the lack of information, or the movies that were never released. I have chosen to eradicate these values during the data cleaning phase.

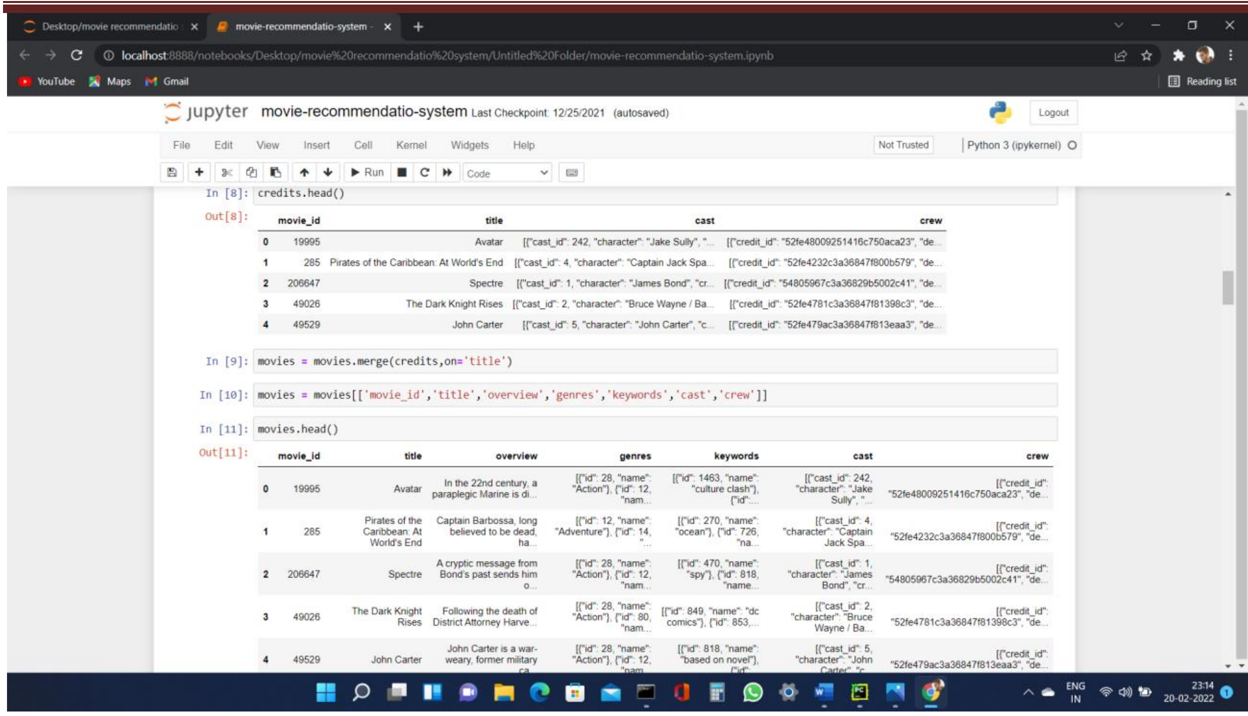
The first column shows the highest profit made by a movie and second column shows the highest in loss movie in this dataset.

As we can see that **'Avatar'** movie Directed by James Cameron earn the highest profit in all, making over 2.5B in profit in this dataset.And the most in loss movie in this dataset is **The Warrior's Way**. Going in loss by more than 400M was directed by Singmoo Lee.

Chapter 7

DATA PREPROCESSING IN JUPYTER NOTEBOOK



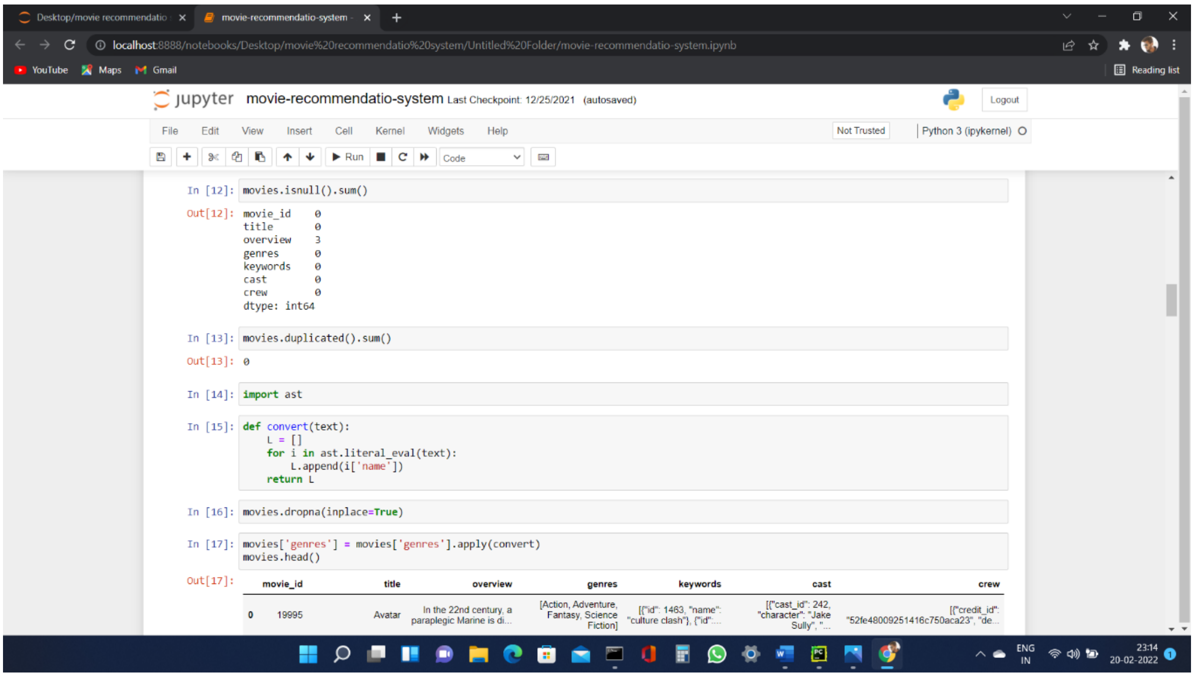


In [8]:
```
credits.head()
```
Out[8]:

	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

In [9]:
```
movies = movies.merge(credits,on='title')
```
In [10]:
```
movies = movies[['movie_id','title','overview','genres','keywords','cast','crew']]
```
In [11]:
```
movies.head()
```
Out[11]:

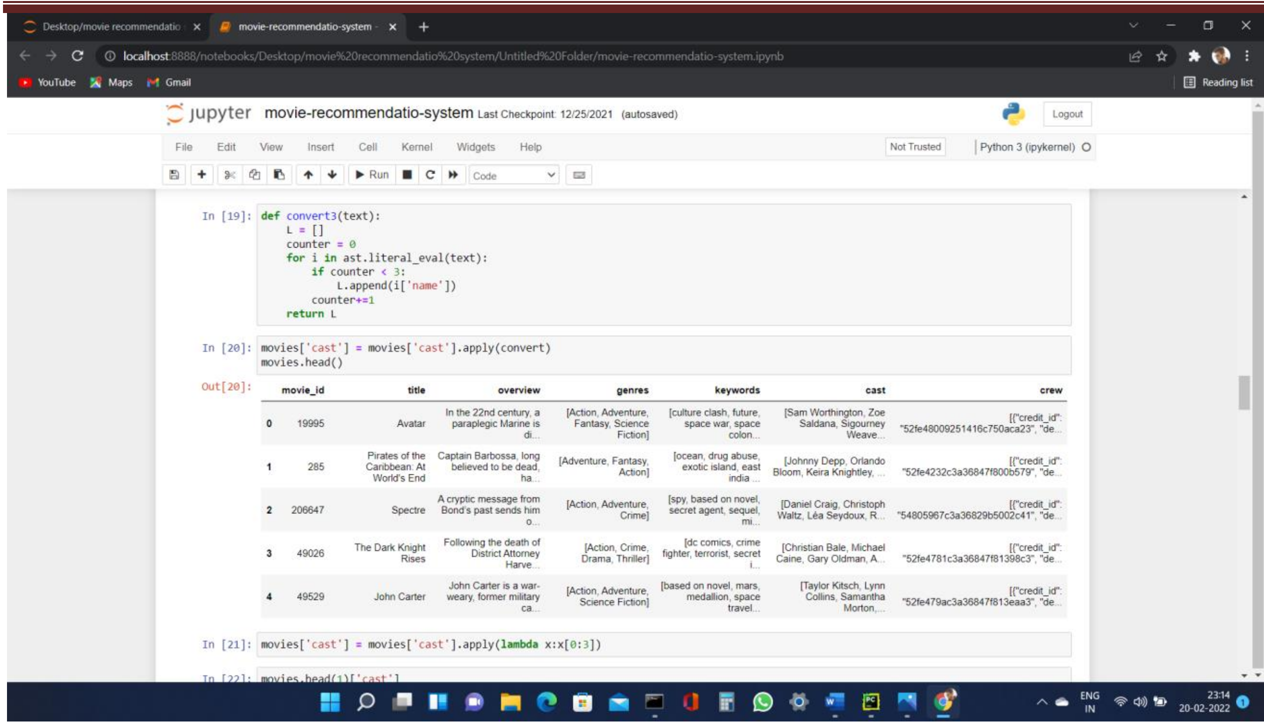
	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...



In [12]:
```
movies.isnull().sum()
```
Out[12]:
```
movie_id    0
title       0
overview    3
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```
In [13]:
```
movies.duplicated().sum()
```
Out[13]: 0

In [14]:
```
import ast
```
In [15]:
```
def convert(text):
    L = []
    for i in ast.literal_eval(text):
        L.append(i['name'])
    return L
```
In [16]:
```
movies.dropna(inplace=True)
```
In [17]:
```
movies['genres'] = movies['genres'].apply(convert)
movies.head()
```
Out[17]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...



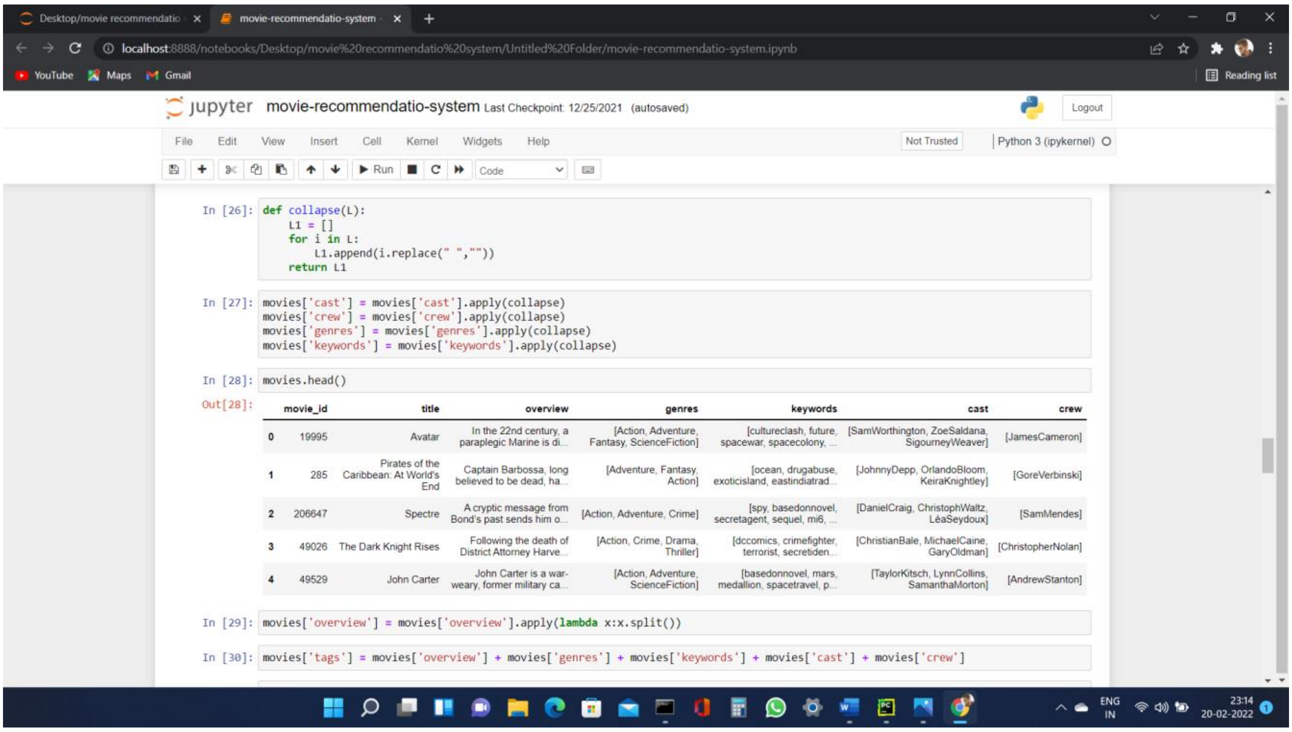
```
In [19]: def convert3(text):
             L = []
             counter = 0
             for i in ast.literal_eval(text):
                 if counter < 3:
                     L.append(i['name'])
                 counter+=1
             return L
```

```
In [20]: movies['cast'] = movies['cast'].apply(convert)
         movies.head()
```

Out[20]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weave...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley, ...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux, R...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman, A...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton,...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

```
In [21]: movies['cast'] = movies['cast'].apply(lambda x:x[0:3])
```



```
In [26]: def collapse(L):
             L1 = []
             for i in L:
                 L1.append(i.replace(" ",""))
             return L1
```

```
In [27]: movies['cast'] = movies['cast'].apply(collapse)
         movies['crew'] = movies['crew'].apply(collapse)
         movies['genres'] = movies['genres'].apply(collapse)
         movies['keywords'] = movies['keywords'].apply(collapse)
```

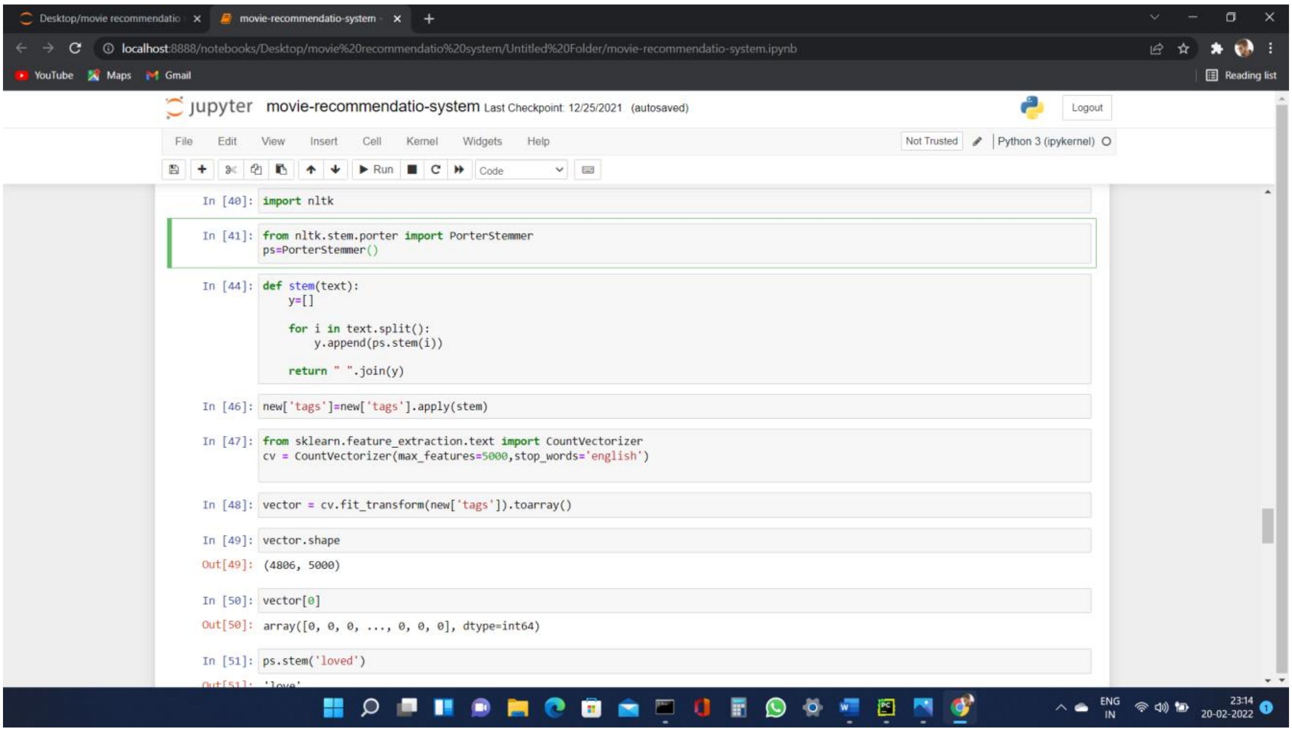
```
In [28]: movies.head()
```

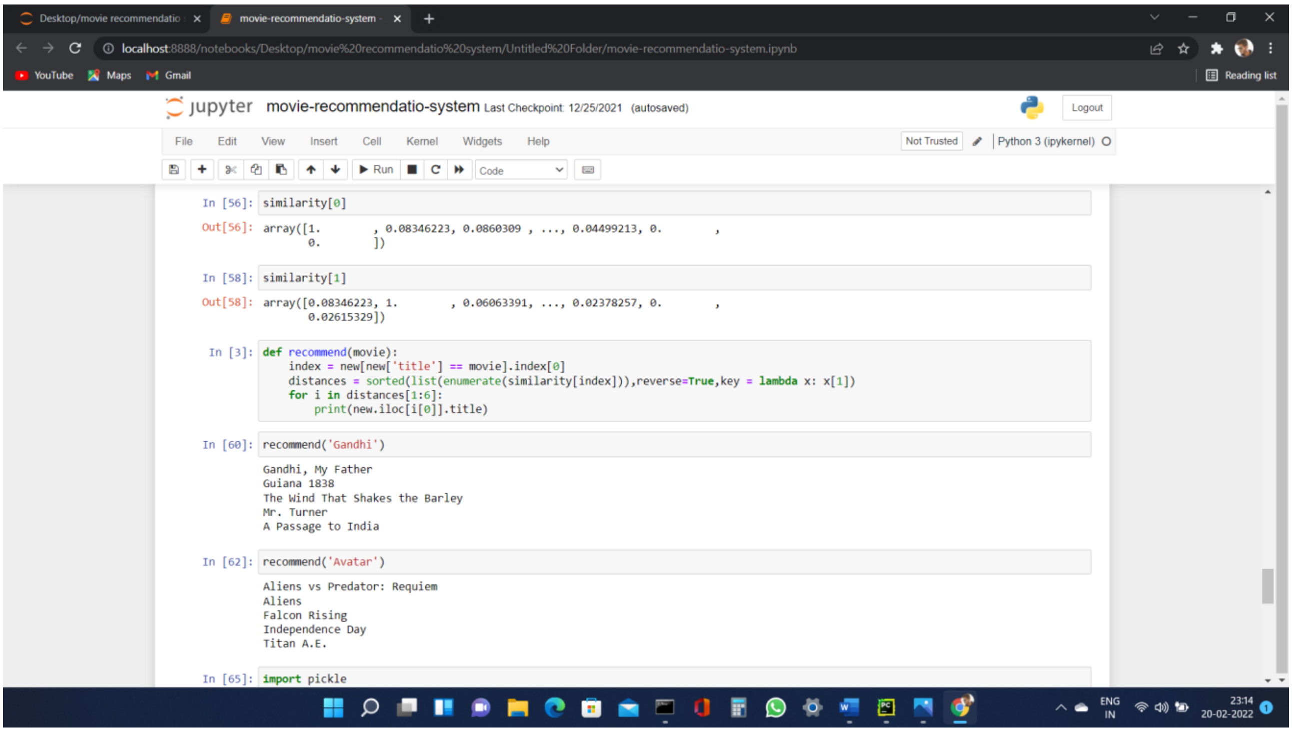
Out[28]:

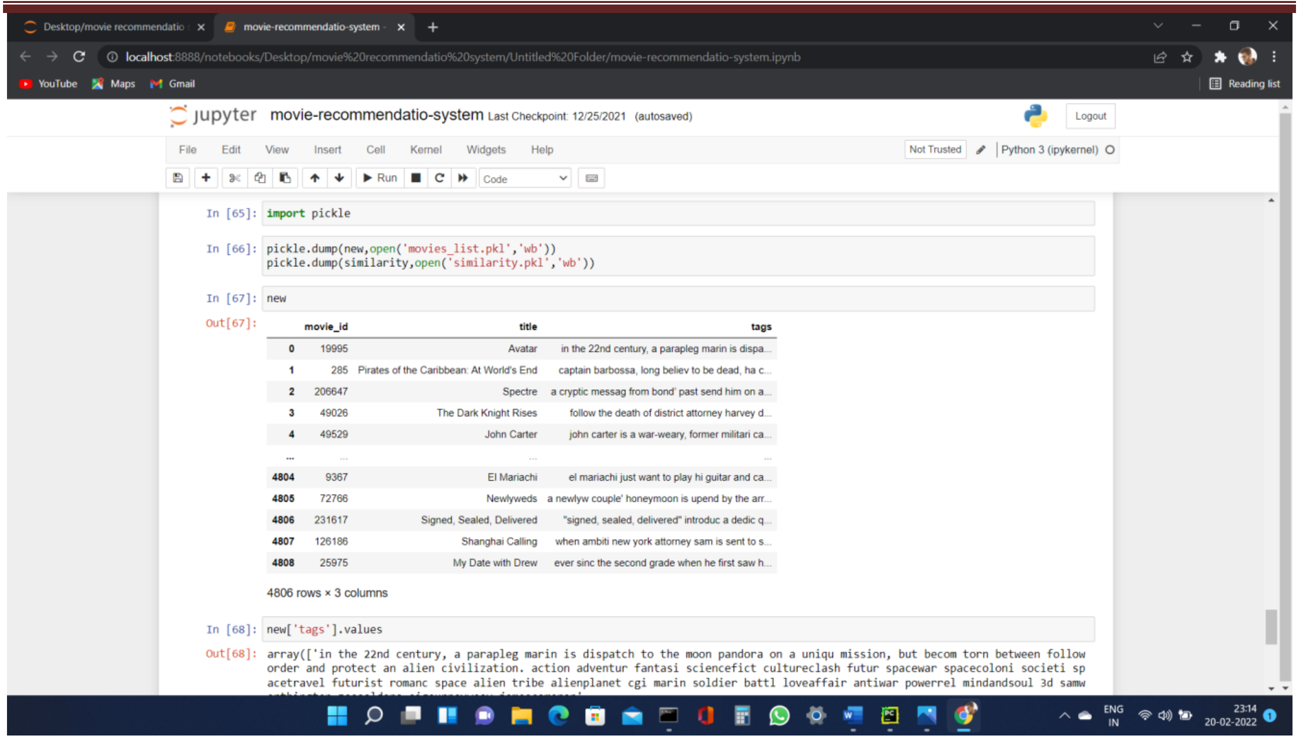
	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]

```
In [29]: movies['overview'] = movies['overview'].apply(lambda x:x.split())
```

```
In [30]: movies['tags'] = movies['overview'] + movies['genres'] + movies['keywords'] + movies['cast'] + movies['crew']
```


jupyter movie-recommendatio-system Last Checkpoint: 12/25/2021 (autosaved)
In [40]: import nltk
In [41]: from nltk.stem.porter import PorterStemmer
ps=PorterStemmer()
In [44]: def stem(text):
y=[]
for i in text.split():
y.append(ps.stem(i))
return " ".join(y)
In [46]: new['tags']=new['tags'].apply(stem)
In [47]: from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=5000,stop_words='english')
In [48]: vector = cv.fit_transform(new['tags']).toarray()
In [49]: vector.shape
Out[49]: (4806, 5000)
In [50]: vector[0]
Out[50]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
In [51]: ps.stem('loved')


jupyter movie-recommendatio-system Last Checkpoint: 12/25/2021 (autosaved)
In [56]: similarity[0]
Out[56]: array([1. , 0.08346223, 0.0860309 , ..., 0.04499213, 0. , 0.])
In [58]: similarity[1]
Out[58]: array([0.08346223, 1. , 0.06063391, ..., 0.02378257, 0. , 0.02615329])
In [3]: def recommend(movie):
index = new[new['title'] == movie].index[0]
distances = sorted(list(enumerate(similarity[index])),reverse=True,key = lambda x: x[1])
for i in distances[1:6]:
print(new.iloc[i[0]].title)
In [60]: recommend('Gandhi')
Gandhi, My Father
Guiana 1838
The Wind That Shakes the Barley
Mr. Turner
A Passage to India
In [62]: recommend('Avatar')
Aliens vs Predator: Requiem
Aliens
Falcon Rising
Independence Day
Titan A.E.
In [65]: import pickle



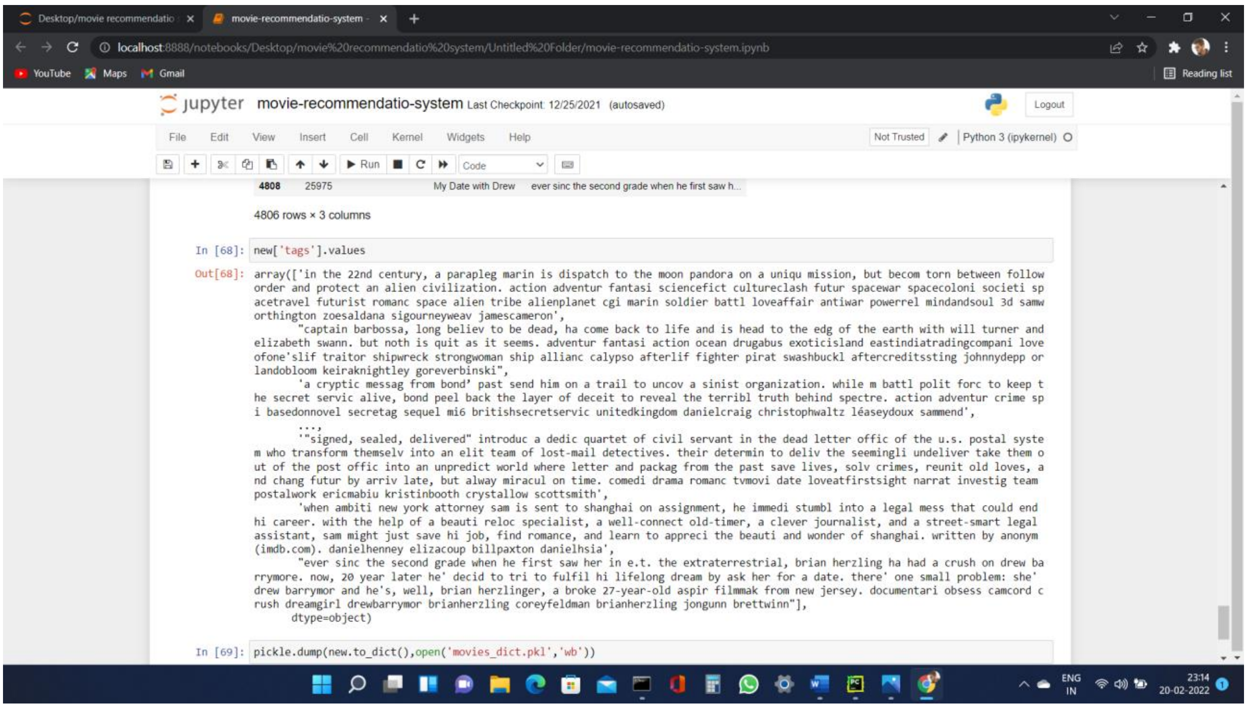
```
In [65]: import pickle

In [66]: pickle.dump(new,open('movies_list.pkl','wb'))
         pickle.dump(similarity,open('similarity.pkl','wb'))

In [67]: new
```

Out[67]:

	movie_id	title	tags
0	19995	Avatar	in the 22nd century, a parapleg marin is dispa...
1	285	Pirates of the Caribbean: At World's End	captain barbossa, long believ to be dead, ha c...
2	206647	Spectre	a cryptic messag from bond' past send him on a...
3	49026	The Dark Knight Rises	follow the death of district attorney harvey d...
4	49529	John Carter	john carter is a war-weary, former militari ca...
...	...	...	...
4804	9367	El Mariachi	el mariachi just want to play hi guitar and ca...
4805	72766	Newlyweds	a newlyw couple' honeymoon is upend by the arr...
4806	231617	Signed, Sealed, Delivered	"signed, sealed, delivered" introduc a dedic q...
4807	126186	Shanghai Calling	when ambiti new york attorney sam is sent to s...
4808	25975	My Date with Drew	ever sinc the second grade when he first saw h...

4806 rows × 3 columns

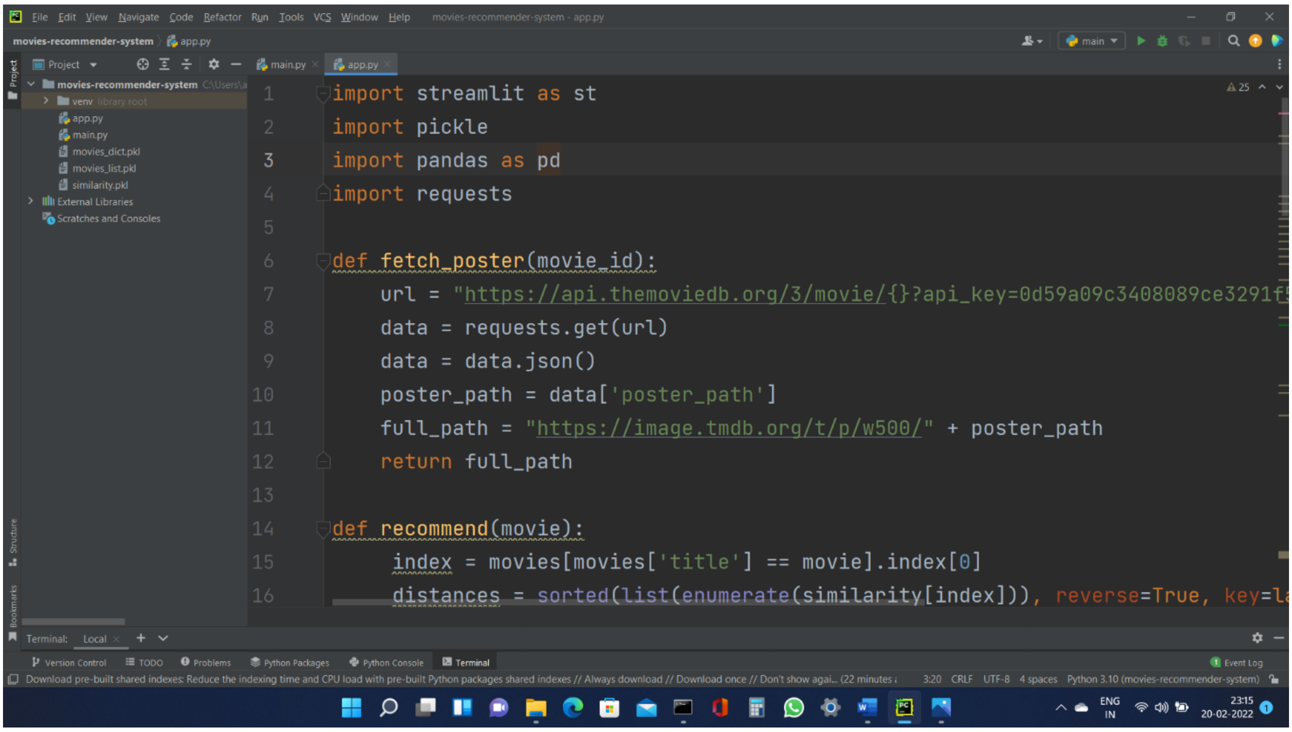
```
In [68]: new['tags'].values

Out[68]: array(['in the 22nd century, a parapleg marin is dispatch to the moon pandora on a uniqu mission, but becom torn between follow
order and protect an alien civilization. action adventur fantasi sciencefict cultureclash futur spacewar spacecoloni societi sp
acetravel futurist romanc space alien tribe alienplanet cgi marin soldier battl loveaffair antiwar powerrel mindandsoul 3d samw
```

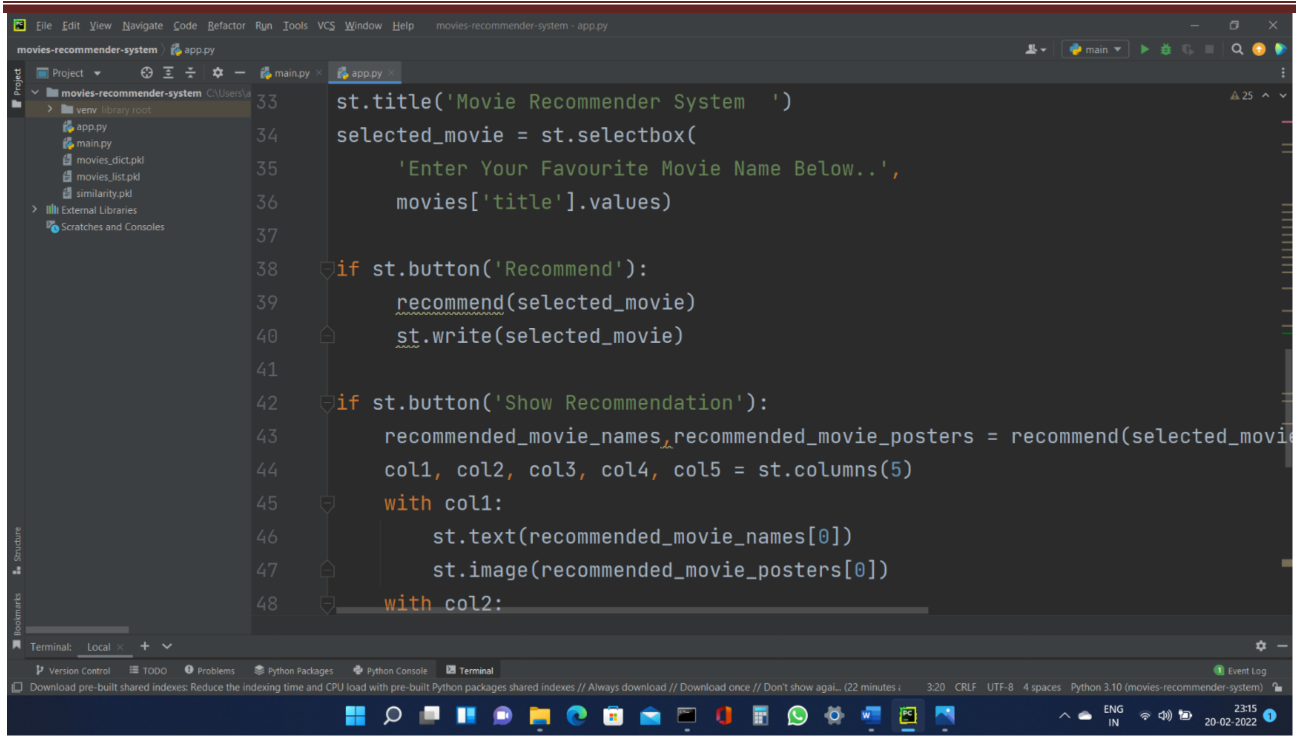


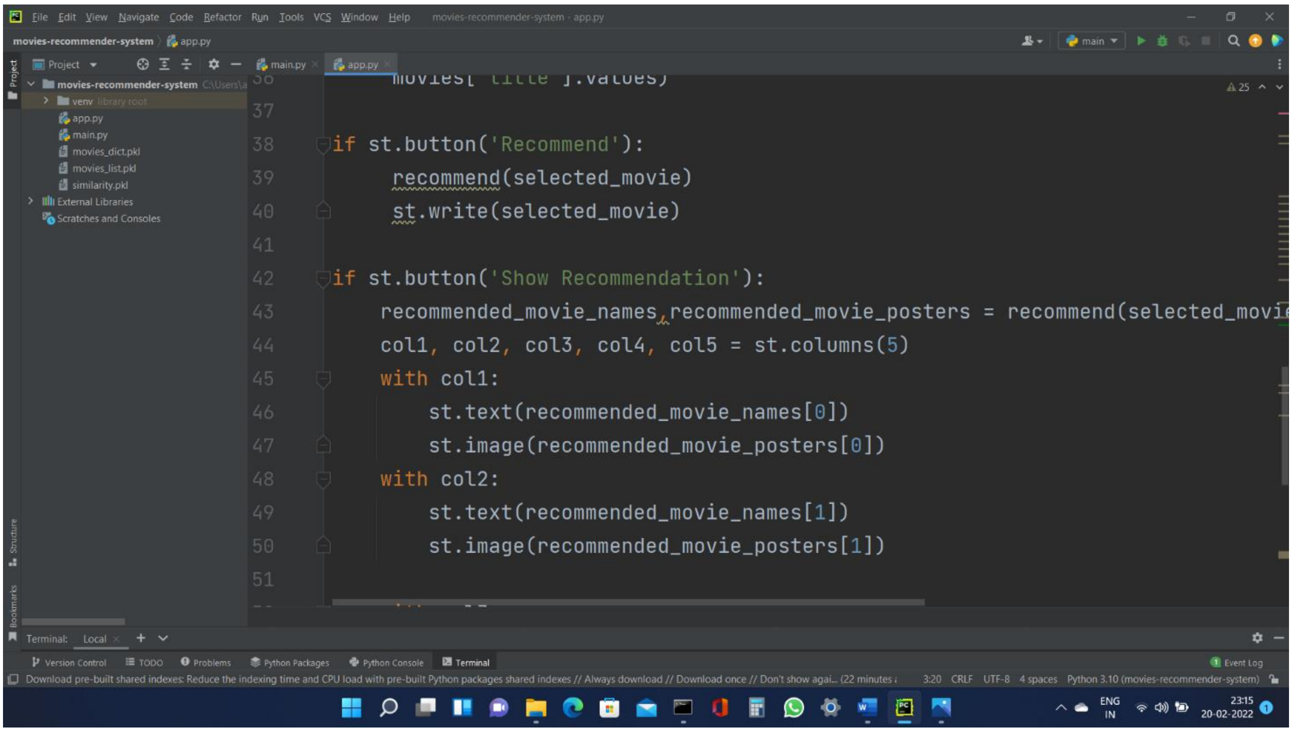
```
4808    25975    My Date with Drew    ever sinc the second grade when he first saw h...

4806 rows × 3 columns

In [68]: new['tags'].values

Out[68]: array(['in the 22nd century, a parapleg marin is dispatch to the moon pandora on a uniqu mission, but becom torn between follow
order and protect an alien civilization. action adventur fantasi sciencefict cultureclash futur spacewar spacecoloni societi sp
acetravel futurist romanc space alien tribe alienplanet cgi marin soldier battl loveaffair antiwar powerrel mindandsoul 3d samw
orthington zoesaldana sigourneyweav jamescameron',
       "captain barbossa, long believ to be dead, ha come back to life and is head to the edg of the earth with will turner and
elizabeth swann. but noth is quit as it seems. adventur fantasi action ocean drugabus exoticisland eastindiatradingcompani love
ofone'slif traitor shipwreck strongwoman ship allianc calypso afterlif fighter pirat swashbuckl aftercreditssting johnnydepp or
landobloom keiraknightley goreverbinski",
       'a cryptic messag from bond' past send him on a trail to uncov a sinist organization. while m battl polit forc to keep t
he secret servic alive, bond peel back the layer of deceit to reveal the terribl truth behind spectre. action adventur crime sp
i basedonnovel secretag sequel mi6 britishsecretservic unitedkingdom danielcraig christophwaltz léaseydoux sammend',
       ...,
       '"signed, sealed, delivered" introduc a dedic quartet of civil servant in the dead letter offic of the u.s. postal syste
m who transform themselv into an elit team of lost-mail detectives. their determin to deliv the seemingli undeliver take them o
ut of the post offic into an unpredict world where letter and packag from the past save lives, solv crimes, reunit old loves, a
nd chang futur by arriv late, but alway miracul on time. comedi drama romanc tvmovi date loveatfirstsight narrat investig team
postalwork ericmabiu kristinbooth crystallow scottsmith',
       'when ambiti new york attorney sam is sent to shanghai on assignment, he immedi stumbl into a legal mess that could end
hi career. with the help of a beauti reloc specialist, a well-connect old-timer, a clever journalist, and a street-smart legal
assistant, sam might just save hi job, find romance, and learn to appreci the beauti and wonder of shanghai. written by anonym
(imdb.com). danielhenney elizacoup billpaxton danielhsia',
       "ever sinc the second grade when he first saw her in e.t. the extraterrestrial, brian herzling ha had a crush on drew ba
rrymore. now, 20 year later he' decid to tri to fulfil hi lifelong dream by ask her for a date. there' one small problem: she'
drew barrymor and he's, well, brian herzlinger, a broke 27-year-old aspir filmmak from new jersey. documentari obsess camcord c
rush dreamgirl drewbarrymor brianherzling coreyfeldman brianherzling jongunn brettwinn"],
      dtype=object)

In [69]: pickle.dump(new.to_dict(),open('movies_dict.pkl','wb'))
```

Chapter 8

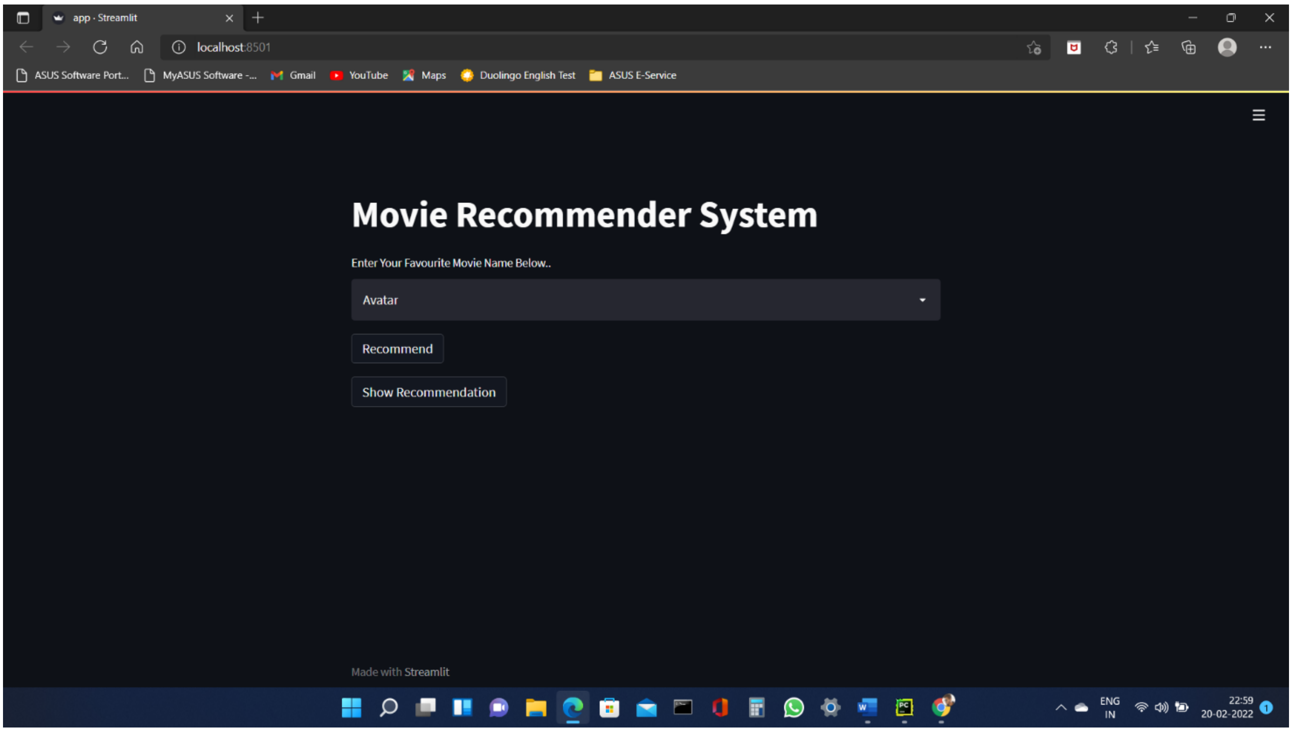
PYTHON

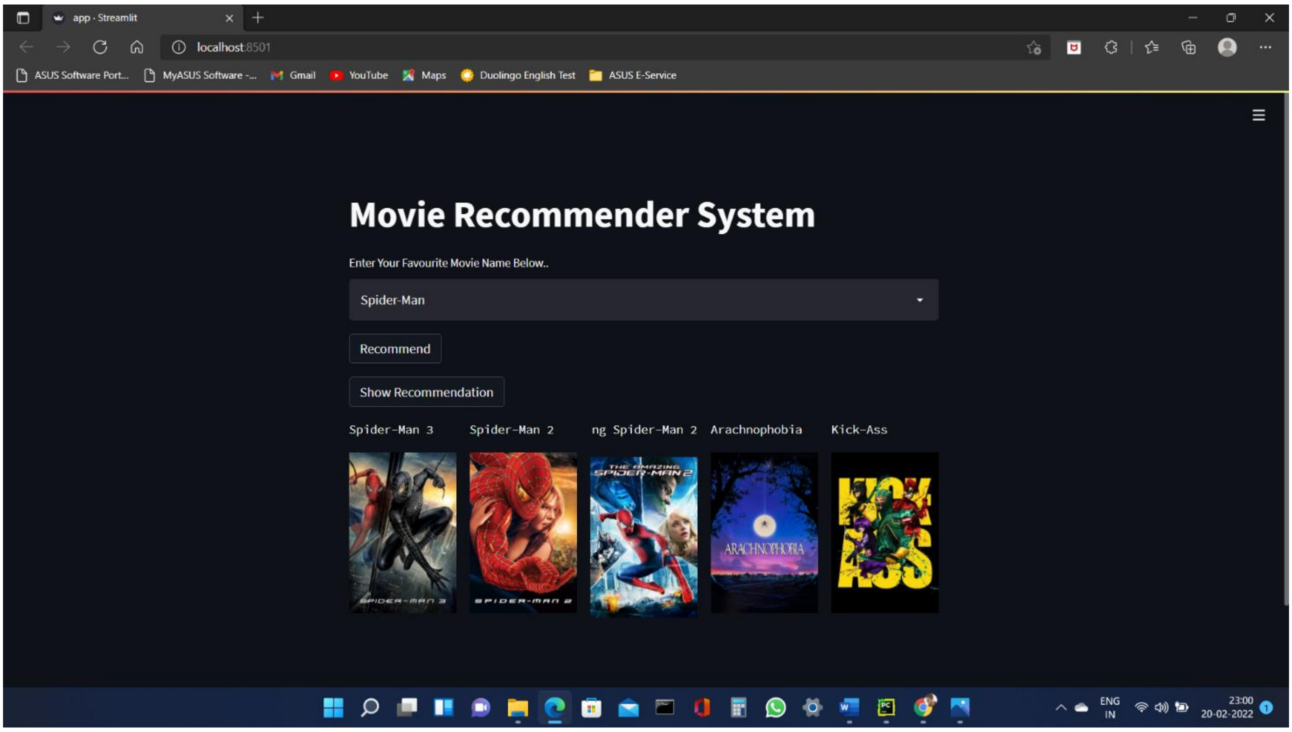


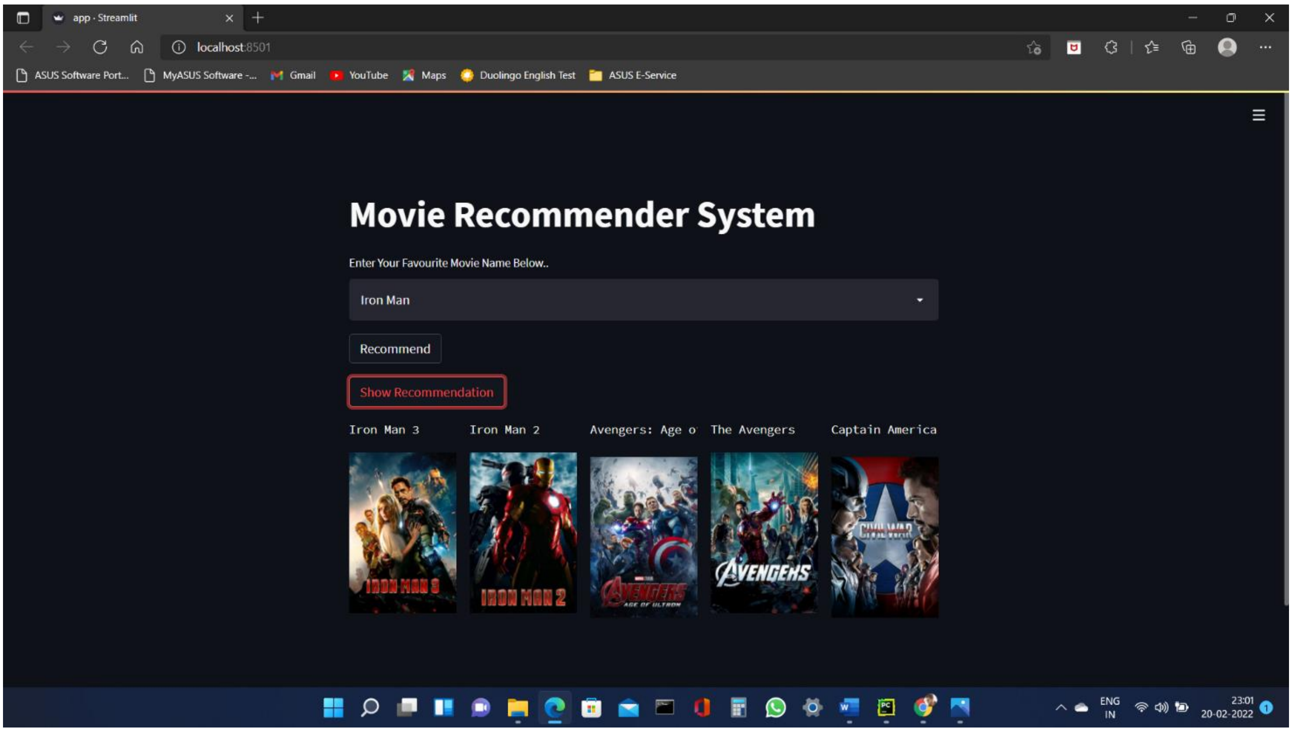

st.title('Movie Recommender System ')
selected_movie = st.selectbox(
'Enter Your Favourite Movie Name Below..',
movies['title'].values)
if st.button('Recommend'):
recommend(selected_movie)
st.write(selected_movie)
if st.button('Show Recommendation'):
recommended_movie_names,recommended_movie_posters = recommend(selected_movi
col1, col2, col3, col4, col5 = st.columns(5)
with col1:
st.text(recommended_movie_names[0])
st.image(recommended_movie_posters[0])
with col2:


if st.button('Recommend'):
recommend(selected_movie)
st.write(selected_movie)
if st.button('Show Recommendation'):
recommended_movie_names,recommended_movie_posters = recommend(selected_movi
col1, col2, col3, col4, col5 = st.columns(5)
with col1:
st.text(recommended_movie_names[0])
st.image(recommended_movie_posters[0])
with col2:
st.text(recommended_movie_names[1])
st.image(recommended_movie_posters[1])

RESULTS


app · Streamlit
localhost:8501
ASUS Software Port...
MyASUS Software -...
Gmail
YouTube
Maps
Duolingo English Test
ASUS E-Service
Movie Recommender System
Enter Your Favourite Movie Name Below..
Avatar
Recommend
Show Recommendation
Made with Streamlit
ENG
IN
22:59
20-02-2022


app · Streamlit
localhost:8501
ASUS Software Port...
MyASUS Software -...
Gmail
YouTube
Maps
Duolingo English Test
ASUS E-Service
Movie Recommender System
Enter Your Favourite Movie Name Below..
Spider-Man
Recommend
Show Recommendation
Spider-Man 3
Spider-Man 2
ng Spider-Man 2
Arachnophobia
Kick-Ass
ENG
IN
23:00
20-02-2022


app · Streamlit
localhost:8501
ASUS Software Port...
MyASUS Software -...
Gmail
YouTube
Maps
Duolingo English Test
ASUS E-Service
Movie Recommender System
Enter Your Favourite Movie Name Below..
Iron Man
Recommend
Show Recommendation
Iron Man 3
Iron Man 2
Avengers: Age o
The Avengers
Captain America
ENG
IN
23:01
20-02-2022

Chapter 10

CONCLUSION

The amount of data available on the internet for different kinds of movies, genre, actor, director is very huge. Therefore, this system is used to consolidate all these data into a representable, user friendly interface where the user can be recommended the best movie of his choice by quick filtering of the information on the web

REFERENCES

YOUTUBE: https://youtu.be/bbObhCQ-c2g

W3SCHOOLS: https://www.w3schools.com/