

Movie Recommendation Model

In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
movies =pd.read_csv(r"C:\Users\Pranav\Desktop\Prachi\Movierecomendation\tmdb_5000_movies.csv")
credits=pd.read_csv(r"C:\Users\Pranav\Desktop\Prachi\Movierecomendation\tmdb_5000_credits.csv")
```

In [3]:

```
movies.shape #count of rows and columns
```

Out[3]:

(4803, 20)

In [4]:

```
credits.shape
```

Out[4]:

(4803, 4)

In [8]:

```
credits.head(1)["crew"].values #this will show the crew details
```

In [6]:

```
movies.head()
```

Out[6]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name Film P...
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"name": "Pictures"
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	[{"name Pictu
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknighttrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	[{"name" Pictures", "i
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	[{"name": "Pictu

In [7]:

```
credits.head()
```

Out[7]:

	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 [8]:

```
cinema= movies.merge(credits,on = "title") #this will merge these two datasets on their common column
```

In [9]:

```
cinema["original_language"].value_counts() #this will show the total number of movies sorted language wise
```

Out[9]:

```
en    4510
fr     70
es     32
zh     27
de     27
hi     19
ja     16
it     14
ko     12
cn     12
ru     11
pt      9
da      7
sv      5
nl      4
fa      4
th      3
he      3
ta      2
cs      2
ro      2
id      2
ar      2
vi      1
sl      1
ps      1
no      1
ky      1
hu      1
pl      1
af      1
nb      1
tr      1
is      1
xx      1
te      1
el      1
Name: original_language, dtype: int64
```

In [10]:

```
cinema.head(1)
```

Out[10]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies	...	ru
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Ingenious Film Partners", "id": ... 289...		

1 rows x 23 columns



In [11]:

```
cinema = cinema[["movie_id","title","overview","genres","keywords","cast","crew"]] #We are taking only the necessary columns according to
```

```
cinema.isnull().sum()
```

```
movie_id    0
title       0
overview    3
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

```
cinema.dropna(inplace=True) #dropping the rows with null values
```

```
cinema.shape
```

(4806, 7)

```
cinema.duplicated().sum() #This will show the duplicate values present if any
```

⑧

```
cinema["genres"][0]
```

```
'[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction}]'
```

```
cinema.head(1)
```

	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", "..."}, {"id": 243, "character": "Trudy", "..."}]}	{{{"credit_id": "52fe48009251416c750aca23"}, {"de..."}}}

```
#
import ast
ast.literal_eval(['{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science"}'])
```

```
[{'id': 28, 'name': 'Action'},
 {'id': 12, 'name': 'Adventure'},
 {'id': 14, 'name': 'Fantasy'},
 {'id': 878, 'name': 'Science Fiction'}]
```

```
["Action", "Adventures", "Fantasy", "Science Fiction"]
```

```
['Action', 'Adventures', 'Fantasy', 'Science Fiction']
```

```
a = eval(input()) #
b = eval(input())
print(a+b)
```

1
1.5
2.5

In [21]:

```
def converted(obj):
    L=[]
    for i in obj:
        L.append(i["name"])
    return L
```

In [22]:

```
import ast
def converted(obj):
    L=[]
    for i in ast.literal_eval(obj):
        L.append(i["name"])
    return L
```

In [23]:

```
# We have created a function converted by using that function in genres we are keeping only the names from dictionary
cinema["genres"] = cinema["genres"].apply(converted)
```

In [24]:

```
cinema["keywords"][0] #As you can see in keywords there are many ditionaries and we only want names
```

Out[24]:

```
'[{"id": 1463, "name": "culture clash"}, {"id": 2964, "name": "future"}, {"id": 3386, "name": "space war"}, {"id": 3388, "name": "space colony"}, {"id": 3679, "name": "society"}, {"id": 3801, "name": "space travel"}, {"id": 9685, "name": "futuristic"}, {"id": 9840, "name": "romance"}, {"id": 9882, "name": "space"}, {"id": 9951, "name": "alien"}, {"id": 10148, "name": "tribe"}, {"id": 10158, "name": "alien planet"}, {"id": 10987, "name": "cgi"}, {"id": 11399, "name": "marine"}, {"id": 13065, "name": "soldier"}, {"id": 14643, "name": "battle"}, {"id": 14720, "name": "love affair"}, {"id": 165431, "name": "anti war"}, {"id": 193554, "name": "power relations"}, {"id": 206690, "name": "mind and soul"}, {"id": 209714, "name": "3d"}]'
```

In [25]:

```
cinema["keywords"] = cinema["keywords"].apply(converted)
```

In [26]:

```
cinema.head(2) #We got only names in genres and keywords hence data looks upto point
```

Out[26]:

	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...	[{"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...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...

In [7]:

```
cinema["cast"][0] #You can see in the cast column in one single row there are many dictionaries of the movie cast members
```

In [28]:

```
#We want only the top three cast members in this case they might be the hero,heroine, villan characters
def convert1(obj):
    L=[]
    counter = 0
    for i in ast.literal_eval(obj):
        if counter !=3:
            L.append(i["name"])
            counter+=1
        else:
            break
    return L
```

In [29]:

```
cinema["cast"] = cinema["cast"].apply(convert1)
```

In [30]:

```
cinema.head(2)
```

Out[30]:

	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 Weaver]	[{"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...

In [1]:

```
cinema["crew"][0] #now in the crew we shall need only the director of the film
```

In [32]:

```
#creating a function for keeping only the directors name
def find_director(obj2):
    L=[]
    for i in ast.literal_eval(obj2):
        if i["job"]=="Director":
            L.append(i["name"])
            break
    return L
```

In [33]:

```
cinema["crew"] = cinema["crew"].apply(find_director)
```

In [34]:

```
cinema.head(2)
```

Out[34]:

	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 Weaver]	[James Cameron]
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]	[Gore Verbinski]

In [35]:

```
cinema["overview"][0]
```

Out[35]:

'In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization.'

In [36]:

```
cinema["overview"] = cinema["overview"].apply(lambda x: x.split()) #We are splitting the overview into single words
```

In [37]:

```
#we are replacting all the spaces in the data to nonspacing
cinema["genres"] = cinema["genres"].apply(lambda x: [i.replace(" ", "") for i in x])
cinema["keywords"] = cinema["keywords"].apply(lambda x:[i.replace(" ", "") for i in x])
cinema["cast"] = cinema["cast"].apply(lambda x:[i.replace(" ", "") for i in x])
cinema["crew"] = cinema["crew"].apply(lambda x:[i.replace(" ", "") for i in x])
```

In [38]:

```
cinema.head(2)
```

Out[38]:

	movie_id		title	overview	genres	keywords	cast	crew
0	19995		Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[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, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]

In [39]:

```
#We are creating a new column by combining all the columns
cinema["tags"] = cinema ["overview"] + cinema ["genres"] +cinema["keywords"] + cinema["cast"] + cinema["crew"]
```

In [40]:

```
cinema.head(2)
```

Out[40]:

	movie_id	title	overview	genres	keywords	cast	crew	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OriandoBloom, KeiraKnightley]	[GoreVerbinski]	[Captain, Barbossa,, long, believed, to, be, d...

In [41]:

```
#Creating a new dataframe consisting of only below mentioned columns
new_cinema = cinema[["movie_id","title","tags"]]
```

In [42]:

```
new_cinema.head(2)
```

Out[42]:

	movie_id	title	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...

In [43]:

```
#We will be using join function as this
a = ["a","b","c"]
" ".join(a)
```

Out[43]:

```
'a b c'
```

In [44]:

```
new_cinema["tags"] = new_cinema["tags"].apply(lambda x:" ".join(x))
```

C:\Users\Pranav\AppData\Local\Temp\ipykernel_12616\1830036848.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
new_cinema["tags"] = new_cinema["tags"].apply(lambda x:" ".join(x))

In [45]:

```
new_cinema["tags"][0]
```

Out[45]:

```
'In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization. Action Adventure Fantasy ScienceFiction cultureclash future spacewar spacecolony society spacetravel futuristic romance space alien tribe alienplanet cgi marine soldier battle loveaffair antiwar powerrelations mindandsoul 3d SamWorthington ZoeSaldana SigourneyWeaver JamesCameron'
```

In [46]:

```
#converting everthing in tags to lower
new_cinema["tags"] = new_cinema["tags"].apply(lambda x: x.lower())
```

C:\Users\Pranav\AppData\Local\Temp\ipykernel_12616\1940561924.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
new_cinema["tags"] = new_cinema["tags"].apply(lambda x: x.lower())

Vectoriztion-converting text into a vector

In [47]:

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 5000, stop_words="english")
```

In [48]:

```
vectors = cv.fit_transform(new_cinema["tags"]).toarray()
```

In [2]:

```
cv.get_feature_names()
```

In [3]:

```
!pip install nltk
```

In [51]:

```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
```

In [52]:

```
ps.stem("Loving") #as you can see the loving, loved is converted to main word Love
```

Out[52]:

```
'love'
```

In [53]:

```
def stem(text):
    y = []
    for i in text.split():
        y.append(ps.stem(i))
    return " ".join(y)
```

In [54]:

```
new_cinema["tags"] = new_cinema["tags"].apply(stem)
```

C:\Users\Pranav\AppData\Local\Temp\ipykernel_12616\498763844.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
new_cinema["tags"] = new_cinema["tags"].apply(stem)

In [4]:

```
# you can see now we dont have any repeative vectors
```

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 5000, stop_words="english")
vectors = cv.fit_transform(new_cinema["tags"]).toarray()
cv.get_feature_names()
```

In [56]:

```
from sklearn.metrics.pairwise import cosine_similarity
```

In [57]:

```
similarity = cosine_similarity(vectors)
```

In [58]:

```
similarity[1]
```

Out[58]:

```
array([0.08346223, 1.          , 0.06063391, ..., 0.02378257, 0.          ,
       0.02615329])
```

In [59]:

```
sorted(similarity[0])[-6:-1]
```

Out[59]:

```
[0.25038669783359574,
 0.255608593705383,
 0.2605130246476754,
 0.26901379342448517,
 0.2867696673382022]
```

In [5]:

```
sorted(similarity[0],reverse=True)
```

In [6]:

```
list(enumerate(similarity[0]))
```

In [62]:

```
sorted(list(enumerate(similarity[0])),reverse=True,key=lambda x:x[1])[1:6]
```

Out[62]:

```
[(1216, 0.2867696673382022),  
(2409, 0.26901379342448517),  
(3730, 0.2605130246476754),  
(507, 0.255608593705383),  
(539, 0.25038669783359574)]
```

In [63]:

```
def recommend(movie):  
    movie_index= new_cinema[new_cinema["title"]==movie].index[0]  
    distances = similarity[movie_index]  
    movie_list = sorted(list(enumerate(distances)),reverse=True,key=lambda x:x[1])[1:6]  
  
    for i in movie_list:  
        print(new_cinema.iloc[i[0]].title)
```

In [64]:

```
recommend("Avatar") #The system is now showing the movies similar to Avatar
```

Aliens vs Predator: Requiem
Aliens
Falcon Rising
Independence Day
Titan A.E.

Type *Markdown* and LaTeX: α^2

In [65]:

```
recommend("The Godfather") #Great!
```

Desert Dancer
Step Up
Take the Lead
Footloose
Tango

In []:

In [66]:

In [67]:

In [68]:

In [69]:

In [70]:

In [71]:

In [85]:

In []:

In [89]:

In []:

In [91]:

In []:

In []: