

Movie Recommender System

```
In [1]: # Importing Required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: # Import the dataset
movies = pd.read_csv("tmdb_5000_movies.csv")
credits = pd.read_csv("tmdb_5000_credits.csv")
```

```
movies.head()
```

Out[3]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id": 1464, "name": "marine"}]	en	Avatar	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 those who have become his family.	150.437577
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}]	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "name": "nautilus"}]	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, has returned. Jack Sparrow, William Turner (Orlando Bloom) and Elizabeth Swann (Keate Winslet) must follow a dangerous path to bring the long lost captain back. But will they save the world or will the dead take over the living?	139.082615
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}]	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}]	en	Spectre	A cryptic message from Bond's old nemesis, the villainous Blofeld, leads him to a secret organization that is plotting to unleash a deadly weapon on the world.	107.376788

```
credits.head()
```

Out[4]:

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": "52fe479ac3a36847f81398c3", "de...

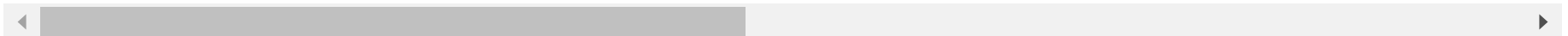
```
In [5]: # Merge the two datasets  
movies = movies.merge(credits, on= 'title')
```

In [6]: movies.head()

Out[6]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	pr
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	
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	I
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	
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	Pi
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	I

5 rows × 23 columns



```
In [7]: # Get some information about the data frame. i.e column names, data types etc.
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4809 entries, 0 to 4808
Data columns (total 23 columns):
#   Column                Non-Null Count  Dtype
---  -
0   budget                4809 non-null   int64
1   genres                4809 non-null   object
2   homepage              1713 non-null   object
3   id                    4809 non-null   int64
4   keywords              4809 non-null   object
5   original_language     4809 non-null   object
6   original_title        4809 non-null   object
7   overview              4806 non-null   object
8   popularity            4809 non-null   float64
9   production_companies  4809 non-null   object
10  production_countries  4809 non-null   object
11  release_date          4808 non-null   object
12  revenue               4809 non-null   int64
13  runtime              4807 non-null   float64
14  spoken_languages      4809 non-null   object
15  status               4809 non-null   object
16  tagline               3965 non-null   object
17  title                4809 non-null   object
18  vote_average          4809 non-null   float64
19  vote_count            4809 non-null   int64
20  movie_id              4809 non-null   int64
21  cast                  4809 non-null   object
22  crew                  4809 non-null   object
dtypes: float64(3), int64(5), object(15)
memory usage: 901.7+ KB
```

```
In [8]: # Keeping required columns only.
movies = movies[['movie_id', 'title', 'overview', 'genres', 'keywords', 'cast', 'crew']]
```

```
In [9]: movies.head()
```

```
Out[9]:
```

	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 [10]: # Check the null values presene in the dataset.  
movies.isnull().sum()
```

```
Out[10]: movie_id      0  
title          0  
overview       3  
genres         0  
keywords       0  
cast           0  
crew           0  
dtype: int64
```

```
In [11]: # drop 3 rows of the column name 'overview'  
movies.dropna(inplace = True)
```

```
In [12]: # Again check the null values presene in the dataset.  
movies.isnull().sum()
```

```
Out[12]: movie_id    0  
         title      0  
         overview   0  
         genres     0  
         keywords   0  
         cast       0  
         crew       0  
         dtype: int64
```

```
In [13]: movies.shape
```

```
Out[13]: (4806, 7)
```

```
In [14]: # Check the duplicate values presene in the dataset.  
movies.duplicated().sum()
```

```
Out[14]: 0
```

```
In [15]: movies.iloc[0].genres
```

```
Out[15]: '[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name":  
"Science Fiction"}]'
```

```
In [16]: import ast
```

```
In [17]: # Define a function to convert 'list of dictionary' into list.
```

```
def gen(x):  
    L = []  
    for i in ast.literal_eval(x):  
        L.append(i['name'])  
    return L
```

```
In [18]: movies['genres'] = movies['genres'].apply(gen)  
movies['genres']
```

```
Out[18]: 0      [Action, Adventure, Fantasy, Science Fiction]  
1      [Adventure, Fantasy, Action]  
2      [Action, Adventure, Crime]  
3      [Action, Crime, Drama, Thriller]  
4      [Action, Adventure, Science Fiction]  
  
      ...  
4804      [Action, Crime, Thriller]  
4805      [Comedy, Romance]  
4806      [Comedy, Drama, Romance, TV Movie]  
4807      []  
4808      [Documentary]  
Name: genres, Length: 4806, dtype: object
```



```
In [19]: movies.head()
```

```
Out[19]:
```

	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...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[{"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...	[Action, Adventure, Crime]	[{"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...	[Action, Crime, Drama, Thriller]	[{"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...	[Action, Adventure, Science Fiction]	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

```
In [20]: # Define a function to convert 'List of dictionary' into List.
```

```
def keywd(x):  
    L = []  
    for i in ast.literal_eval(x):  
        L.append(i['name'])  
    return L
```

```
In [21]: movies['keywords'] = movies['keywords'].apply(keywd)
movies['keywords']
```

```
Out[21]: 0      [culture clash, future, space war, space colon...
1      [ocean, drug abuse, exotic island, east india ...
2      [spy, based on novel, secret agent, sequel, mi...
3      [dc comics, crime fighter, terrorist, secret i...
4      [based on novel, mars, medallion, space travel...

...
4804    [united states-mexico barrier, legs, arms, pap...
4805    []
4806    [date, love at first sight, narration, investi...
4807    []
4808    [obsession, camcorder, crush, dream girl]
Name: keywords, Length: 4806, dtype: object
```

```
In [22]: # Define a function to convert 'list of dictionary' into list.
# Apply this on first 3 dictionaries.
```

```
def cast(x):
    L = []
    stopper = 0
    for i in ast.literal_eval(x):
        if stopper != 3:
            L.append(i['name'])
            stopper+= 1
        else:
            break
    return L
```

```
In [23]: movies['cast'] = movies['cast'].apply(cast)
movies['cast']
```

```
Out[23]: 0      [Sam Worthington, Zoe Saldana, Sigourney Weaver]
1      [Johnny Depp, Orlando Bloom, Keira Knightley]
2      [Daniel Craig, Christoph Waltz, Léa Seydoux]
3      [Christian Bale, Michael Caine, Gary Oldman]
4      [Taylor Kitsch, Lynn Collins, Samantha Morton]

...

4804   [Carlos Gallardo, Jaime de Hoyos, Peter Marqua...
4805   [Edward Burns, Kerry Bishé, Marsha Dietlein]
4806   [Eric Mabiús, Kristin Booth, Crystal Lowe]
4807   [Daniel Henney, Eliza Coupe, Bill Paxton]
4808   [Drew Barrymore, Brian Herzlinger, Corey Feldman]
Name: cast, Length: 4806, dtype: object
```

```
In [24]: # Define a function to convert 'list of dictionary' into list.
# Extract only director names.
```

```
def dirtr(x):
    L = []
    for i in ast.literal_eval(x):
        if i['job'] == 'Director':
            L.append(i['name'])
            break
    return L
```

```
In [25]: movies['crew'] = movies['crew'].apply(dirtr)
movies['crew']
```

```
Out[25]: 0      [James Cameron]
1      [Gore Verbinski]
2      [Sam Mendes]
3      [Christopher Nolan]
4      [Andrew Stanton]
...
4804   [Robert Rodriguez]
4805   [Edward Burns]
4806   [Scott Smith]
4807   [Daniel Hsia]
4808   [Brian Herzlinger]
Name: crew, Length: 4806, dtype: object
```

```
In [26]: movies.head()
```

```
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...	[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]
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]	[Sam Mendes]
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]	[Christopher Nolan]
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]	[Andrew Stanton]

```
In [27]: movies['overview']
```

```
Out[27]: 0      In the 22nd century, a paraplegic Marine is di...
          1      Captain Barbossa, long believed to be dead, ha...
          2      A cryptic message from Bond's past sends him o...
          3      Following the death of District Attorney Harve...
          4      John Carter is a war-weary, former military ca...

          ...

          4804     El Mariachi just wants to play his guitar and ...
          4805     A newlywed couple's honeymoon is upended by th...
          4806     "Signed, Sealed, Delivered" introduces a dedic...
          4807     When ambitious New York attorney Sam is sent t...
          4808     Ever since the second grade when he first saw ...
          Name: overview, Length: 4806, dtype: object
```

```
In [28]: # Convert the sentence into list for overview column.
          movies['overview'] = movies['overview'].apply(lambda x:x.split())
          movies['overview']
```

```
Out[28]: 0      [In, the, 22nd, century,, a, paraplegic, Marin...
          1      [Captain, Barbossa,, long, believed, to, be, d...
          2      [A, cryptic, message, from, Bond's, past, send...
          3      [Following, the, death, of, District, Attorney...
          4      [John, Carter, is, a, war-weary,, former, mili...

          ...

          4804     [El, Mariachi, just, wants, to, play, his, gui...
          4805     [A, newlywed, couple's, honeymoon, is, upended...
          4806     ["Signed,, Sealed,, Delivered", introduces, a,...
          4807     [When, ambitious, New, York, attorney, Sam, is...
          4808     [Ever, since, the, second, grade, when, he, fi...
          Name: overview, Length: 4806, dtype: object
```

```
In [29]: movies.head(2)
```

```
Out[29]:
```

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[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, d...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley]	[Gore Verbinski]

```
In [30]: # Delete the space between two words.
```

```
movies['genres'] = movies['genres'].apply(lambda x: [i.replace(" ", "") for i in x])
movies['genres']
```

```
Out[30]: 0      [Action, Adventure, Fantasy, ScienceFiction]
1      [Adventure, Fantasy, Action]
2      [Action, Adventure, Crime]
3      [Action, Crime, Drama, Thriller]
4      [Action, Adventure, ScienceFiction]
...
4804      [Action, Crime, Thriller]
4805      [Comedy, Romance]
4806      [Comedy, Drama, Romance, TVMovie]
4807      []
4808      [Documentary]
Name: genres, Length: 4806, dtype: object
```

```
In [31]: movies['keywords'] = movies['keywords'].apply(lambda x: [i.replace(" ", "") for i in x])
movies['cast'] = movies['cast'].apply(lambda x: [i.replace(" ", "") for i in x])
movies['crew'] = movies['crew'].apply(lambda x: [i.replace(" ", "") for i in x])
```

```
In [32]: movies.head(2)
```

```
Out[32]:
```

	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 [33]: # Create a new column combining 4 columns i.e genres,keywords,cast & crew.
movies['tags'] = movies['overview'] + movies['genres'] + movies['keywords']+movies['cast'] + movies['crew']
```

```
In [34]: # Create a new dataframe.
df = movies [['movie_id','title','tags']]
df.head()
```

```
Out[34]:
```

	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...
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...

```
In [35]: # Convert list into string by joining the space.
df['tags'] = df['tags'].apply(lambda x: " ".join(x))
df['tags']
```

```
Out[35]: 0      In the 22nd century, a paraplegic Marine is di...
1      Captain Barbossa, long believed to be dead, ha...
2      A cryptic message from Bond's past sends him o...
3      Following the death of District Attorney Harve...
4      John Carter is a war-weary, former military ca...

...
4804    El Mariachi just wants to play his guitar and ...
4805    A newlywed couple's honeymoon is upended by th...
4806    "Signed, Sealed, Delivered" introduces a dedic...
4807    When ambitious New York attorney Sam is sent t...
4808    Ever since the second grade when he first saw ...
Name: tags, Length: 4806, dtype: object
```

```
In [36]: df.head()
```

```
Out[36]:
```

	movie_id	title	tags
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...
2	206647	Spectre	A cryptic message from Bond's past sends him o...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...
4	49529	John Carter	John Carter is a war-weary, former military ca...

```
In [37]: # Convert all the content present in the tags column to lower case.
df['tags'] = df['tags'].apply(lambda x:x.lower())
```



```
In [38]: # convert the tags column into vector.  
from sklearn.feature_extraction.text import CountVectorizer  
cv = CountVectorizer (max_features = 5000, stop_words= 'english')
```

```
In [39]: vectors = cv.fit_transform(df['tags']).toarray()  
vectors
```

```
Out[39]: array([[0, 0, 0, ..., 0, 0, 0],  
               [0, 0, 0, ..., 0, 0, 0],  
               [0, 0, 0, ..., 0, 0, 0],  
               ...,  
               [0, 0, 0, ..., 0, 0, 0],  
               [0, 0, 0, ..., 0, 0, 0],  
               [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

```
In [40]: cv.get_feature_names()
```

```
Out[40]: ['000',  
          '007',  
          '10',  
          '100',  
          '11',  
          '12',  
          '13',  
          '14',  
          '15',  
          '16',  
          '17',  
          '18',  
          '18th',  
          '19',  
          '1930s',  
          '1940s',  
          '1950',  
          '1950s',  
          '1960s',  
          '1970s']
```

```
In [41]: # To combine similar words like actor, actors, acting etc.
import nltk
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
```

```
In [42]: def stem(x):
    L = []
    for i in x.split():
        L.append(ps.stem(i))
    return " ".join(L)
```

```
In [43]: df['tags'] = df['tags'].apply(stem)
df['tags']
```

```
Out[43]: 0      in the 22nd century, a parapleg marin is dispa...
1      captain barbossa, long believ to be dead, ha c...
2      a cryptic messag from bond' past send him on a...
3      follow the death of district attorney harvey d...
4      john carter is a war-weary, former militari ca...

      ...
4804    el mariachi just want to play hi guitar and ca...
4805    a newlyw couple' honeymoon is upend by the arr...
4806    "signed, sealed, delivered" introduc a dedic q...
4807    when ambiti new york attorney sam is sent to s...
4808    ever sinc the second grade when he first saw h...
Name: tags, Length: 4806, dtype: object
```

```
In [44]: # Need to define similarity between 2 tags or movies.
from sklearn.metrics.pairwise import cosine_similarity
```

```
In [45]: similarity = cosine_similarity(vectors)
```

```
In [46]: similarity[0]
```

```
Out[46]: array([1.          , 0.08964215, 0.05976143, ..., 0.02519763, 0.02817181,
                0.          ])
```

```
In [47]: # Make a list of similar movies for index no '0' and sort them.
sorted_list = sorted(list(enumerate(similarity[0])), reverse = True, key=lambda x:x[1])[1:6]
sorted_list
```

```
Out[47]: [(539, 0.26089696604360174),
(1194, 0.2581988897471611),
(507, 0.25302403842552984),
(260, 0.25110592822973776),
(1216, 0.24944382578492943)]
```

```
In [48]: # Define a function for recommendation.
def recommend(movie):
    movie_index = df[df['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(df.iloc[i[0]].title)
```

```
In [51]: recommend('The Dark Knight Rises')
```

```
The Dark Knight
Batman Begins
Batman
Batman Returns
Batman
```

Conclusion:

The movie recommendation system is a valuable tool for providing personalized movie recommendations to customers. By leveraging user data and utilizing advanced algorithms, the system can accurately recommend movies as per user preferences. As the system continues to evolve, it is likely to become even more accurate and efficient. This will help movie lovers find the perfect movie to watch, while also helping movie producers and distributors reach their target audiences.

