In [4]:	<pre>import pandas as pd</pre>
In [5]:	<pre>import numpy as np movies = pd.read_csv('tmdb_5000_credits.csv') credits = pd.read_csv('tmdb_5000_movies.csv')</pre>
In [6]:	movies.head(2)
Out[6]:	movie_idtitlecast019995Avatar[{"cast_id": 242, "character": "Jake Sully", "[{"credit_id": "52fe48009251416c750aca23", "de1285Pirates of the Caribbean: At World's End[{"cast_id": 4, "character": "Captain Jack Spa[{"credit_id": "52fe4232c3a36847f800b579", "de
<pre>In [7]: Out[7]:</pre>	movies.shape (4803, 4)
In [8]: Out[8]:	budget genres homepage id keywords original_language original_title overview popularity production_companies production_countries release_date revenue runtime sports to the contract of the c
	[{"id": 28,
	Captain Barbossa, long believed to be dead, "id": 14, " " [("id": 12, "iname": "Name": "
	ha A cryptic [{"id": 28,
	"nam "name sends him o [{"id": 28,
	\[\begin{array}{cccccccccccccccccccccccccccccccccccc
	4 260000000 "Action"}, http://movies.disney.com/john-carter 49529 "based on en John Carter weary, 43.926995 [{"name": "Walt Disney "name": "United States 2012-03-07 284139100 132.0 [{" name": "Walt Disney "name": "United States 2012-03-07 284139100 132.0 [" name": "United States 2012-03-07 284139100 132.0 "r on the state of the
In [9]: In [10]:	<pre>movies = movies.merge(credits, on='title') movies.head() # budget</pre>
	<pre># homepage # id # original_language # original_title # popularity</pre> # production_company
Out[10]:	# production_comapny # production_countries # release-date(not sure) movie_id title cast crew budget genres homepage id keywords original_language production_companies production_countries release_date(not sure)
	["id": 28, 242, [startid": 28, 242, startid": "name": "name": "name": "startid": 28, startid: "52fe48009251416c750aca23", 237000000 "Action"}, http://www.avatarmovie.com/ 1995 "Sully", " starting in the
	Pirates of the 4, [{"credit_id":
	[{"cast_id": [{"cast_id": [{"id": 28, [{"id": 470, [{"cast_id": "name": "Columbia [{"iso_3166_1": "GB", "name": "James Bond", "cr "cr "cr "cr "cr "cast_id": [{"id": 28, [{"id": 470, "name":
	[{"id": 28,
	["id": 28,
In [11]:	c 5 rows × 23 columns movies = movies[['movie_id', 'title', 'overview', 'genres', 'keywords', 'cast', 'crew']]
In [12]:	movies = movies[['movie_id','title','overview','genres','keywords','cast','crew']] movies.head() movie_id title overview genres keywords cast crew
Out[12]:	movie_idtitleoverviewgenreskeywordscastcrew019995AvatarIn the 22nd century, a paraplegic Marine is di[{"id": 28, "name": "Action"}, {"id": 12, "name": "Culture clash"}, "name": "culture clash"}, "id":[{"cast_id": 242, "character": "Jake Sully", ""52fe48009251416c750aca23", "de1285Pirates of the Caribbean: At World's EndCaptain Barbossa, long believed to be dead, ha[{"id": 12, "name": "Adventure"}, {"id": 1463, "name": "culture clash"}, ["id": 270, "name": "ocean"}, {"id": 726, "na[{"cast_id": 4, "character": "Captain Jack Spa"52fe4232c3a36847f800b579", "de
	2 206647 Spectre Spectra Spectra Spectro Spectra Spectra Spectro Spectra Spectra Spectro Spect
In [15]:	4 49529 John Carter John Carter is a war-weary, former military ca John Carter is a war-weary, former military ca #removing null values movies.isnull().sum() John Carter John Carter is a war-weary, former military ca ["id": 28, "name": "Action"}, {"id": 12, ["id": 818, "name": "based on novel"}, ["id": 5, "character": "John Carter", "c "52fe479ac3a36847f813eaa3", "de "52fe479ac3a36847f81aeaa3", "de "52fe479ac3a36847f81aeaa3", "de "52fe479ac3a36847f81aeaa3", "de "52fe479ac3a36847f81aeaa3", "de "52fe479ac3a36847f81aeaa3", "de "52fe479ac3a36847f81aeaa3", "de "52fe479a
out[15].	movies.isnull().sum() movie_id 0 title 0 overview 0 genres 0
	keywords 0 cast 0 crew 0 dtype: int64
In [19]:	<pre>movies.dropna(inplace=True) movies.duplicated().sum() 0</pre>
Out[19]: In [20]:	import ast
In [21]:	<pre>#getting name from dictionary def convert(text): L = [] for i in ast.literal_eval(text): L.append(i['name'])</pre>
In [22]:	<pre>return L movies['genres'] = movies['genres'].apply(convert) movies.head()</pre>
Out[22]:	movie_id title overview genres keywords cast crew 1995 Avatar In the 22nd century, a paraplegic Marine is di [Action, Adventure, Fantasy, Science Fiction] [{"id": 1463, "name": "culture clash"}, {"id": Sully", " Sully", " Sully", " Sully", " [{"cast_id": 242, "character": "Jake Sully", " Sully", " "52fe48009251416c750aca23", "de [{"credit_id": 270, "name": "ocean"}, {"id": 270, "name": "ocean"}, {"id": 4, "character": "Captain [{"credit_id": 270, "name": "ocean"}, {"id": 270, "name": "coean"}, {"id": 270, "name": "cast_id": 4, "character": "Captain [{"credit_id": 270, "name": "coean"}, {"id": 270, "name": "coean"}, {"coean"}, {"id": 270, "name": "coean"}, {"id":
	Pirates of the Caribbean: At World's End Captain Barbossa, long believed to be dead, ha [Adventure, Fantasy, Action] [Ad
In [23]:	Harve Harve John Carter is a war-weary, former military ca Mayne / Ba [Action, Crime, Drama, Thinler] B53, Wayne / Ba [Ste4781c3a36847f81398c3", "de [Cid": 818, "name": "based on novel"},
In [23]: Out[23]:	movies['keywords'] = movies['keywords'].apply(convert) movies.head() movie_id title overview genres keywords cast crew Avatar In the 22nd century, a paraplegic Marine [Action, Adventure, Fantasy, Science [culture clash, future, space war, space [f"cast_id": 242, "character": "Jake [f"credit_id":
	Fiction] colon Sully", " "52fe48009251416c750aca23", "de 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 "52fe4232c3a36847f800b579", "de A cryptic message from Bond's past [Action Adventure Crimal Space on novel, secret agent, ["cast_id": 1, "character": "James ["credit_id": 1, "character": "James ["credit_id": 2, "character": "James ["cast_id": 1, "character": "James ["credit_id": 2, "character": "James ["credit_i
	2 206647 Spectre Spectro Spectro Spectro Spectre Spectro Spect
In [24]:	<pre>import ast ast.literal_eval('[{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]')</pre>
Out[24]:	<pre>[{'id': 28, 'name': 'Action'}, {'id': 12, 'name': 'Adventure'}, {'id': 14, 'name': 'Fantasy'}, {'id': 878, 'name': 'Science Fiction'}]</pre> <pre> def convert3(text):</pre>
	<pre>L = [] counter = 0 for i in ast.literal_eval(text): if counter < 3: L.append(i['name'])</pre>
In [26]:	<pre>counter+=1 return L movies['cast'] = movies['cast'].apply(convert) movies.head()</pre>
Out[26]:	movies head() movie_id title overview genres keywords cast crew 1995 Avatar In the 22nd century, a paraplegic Marine is di [Action, Adventure, Fantasy, Science Fiction] [Culture clash, future, space war, space colon Sigourney Weave "52fe48009251416c750aca23", "de "52fe48009251416c7
	Pirates of the Caribbean: At World's End Captain Barbossa, long believed to be dead, ha [Adventure, Fantasy, Action] [Adventure, Fantasy, Action] [Spy, based on novel, secret agent, sequel, mi [Daniel Craig, Christoph Waltz, Léa Seydoux, R [Captain Barbossa, long believed to be dead, ha [Captain Barbossa, long
	The Dark Knight Rises
In [27]: In [28]:	<pre>movies['cast'] = movies['cast'].apply(lambda x:x[0:3]) def fetch_director(text): L = []</pre>
	<pre>for i in ast.literal_eval(text): if i['job'] == 'Director': L.append(i['name']) return L</pre>
In [29]: In [30]:	<pre>movies['crew'] = movies['crew'].apply(fetch_director) #movies['overview'] = movies['overview'].apply(lambda x:x.split()) movies.sample(5)</pre>
Out[30]:	movies sample (5) movie_id title overview genres keywords cast crew 305 9016 Treasure Planet When space galleon cabin boy Jim Hawkins disco [Adventure, Animation, Family, Fantasy, Scienc [Cyborg, based on novel, space marine, mutiny, [Joseph Gordon-Levitt, Brian Murray, David Hyd [Ron Clements, John Musker]
	Pete and Debbie are both about to turn 40, the This Is 40 Pete and Debbie are both about to turn 40, the [Comedy] [wife husband relationship, children, family r [Paul Rudd, Leslie Mann, Jason Segel] [Judd Apatow] The true story of top student and athlete, Chr [Adventure, Drama] [male nudity, parents kids relationship, campi [Emile Hirsch, Marcia Gay Harden, William Hurt] [Sean Penn]
	Transylvania Being Julia Lambert is a true diva: beautiful, talen [Animation, Comedy, Family, Fantasy] [Witch, magic, mummy, vampire, dracula, skelet] [Adam Sandler, Steve Buscemi, David Spade] [Genndy Tartakovsky] [István Szabó]
In [31]:	<pre>#removing space def collapse(L): L1 = [] for i in L: L1.append(i.replace(" ", ""))</pre>
In [32]:	<pre>movies['cast'] = movies['cast'].apply(collapse) movies['crew'] = movies['crew'].apply(collapse)</pre>
In [33]:	<pre>movies['Crew'] = movies['Crew'].apply(collapse) movies['genres'] = movies['keywords'].apply(collapse) movies['keywords'] = movies['keywords'].apply(collapse)</pre> movies.head()
Out[33]:	movie_id title overview genres keywords cast crew 1995 Avatar In the 22nd century, a paraplegic Marine is di [Action, Adventure, Fantasy, ScienceFiction] [Cultureclash, future, spacewar, spacecolony, ScienceFiction] [SamWorthington, ZoeSaldana, SigourneyWeaver] [JamesCameron]
	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 Spectre Spectre The Dark Knight Rises Following the death of District Attorney Harve [Action, Crime, Drama, Thriller] [dccomics, crimefighter, terrorist, sequel, mi6, [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 [34]: In [35]:	<pre>movies['overview'] = movies['overview'].apply(lambda x:x.split()) movies['tags'] = movies['overview'] + movies['genres'] + movies['keywords'] + movies['crew']</pre>
In [36]:	<pre>new = movies.drop(columns=['overview', 'genres', 'keywords', 'cast', 'crew']) #new.head() new['tags'] = new['tags'].apply(lambda x: " ".join(x))</pre>
Out[37]:	new['tags'] = new['tags'].apply(lambda x: " ".join(x)) new.head() movie_id title tags 1 9995 Avatar In the 22nd century, a paraplegic Marine is di
	1285Pirates of the Caribbean: At World's EndCaptain Barbossa, long believed to be dead, ha2206647SpectreA cryptic message from Bond's past sends him o349026The Dark Knight RisesFollowing the death of District Attorney Harve
In [38]:	4 49529 John Carter John Carter is a war-weary, former military ca from sklearn.feature_extraction.text import CountVectorizer cv = CountVectorizer(max_features=5000, stop_words='english')
In [39]: In [40]:	<pre>vector = cv.fit_transform(new['tags']).toarray() vector.shape</pre>
In [40]: Out[40]: In [41]:	vector.shape (4806, 5000) from sklearn.metrics.pairwise import cosine_similarity
In [42]:	<pre>from sklearn.metrics.pairwise import cosine_similarity</pre> similarity = cosine_similarity(vector)
In [43]: Out[43]:	similarity array([[1.
	0.], [0.06071767, 0.06350006, 1. ,, 0.02677398, 0. , 0.],, [0.02519763, 0.02635231, 0.02677398,, 1. , 0.07352146,
	[0.02519703, 0.02037398,, 1.
In [44]: Out[44]:	<pre>new[new['title'] == 'The Lego Movie'].index[0] 744</pre>
In [45]:	<pre>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)</pre>
In [46]:	recommend('Gandhi')
	Gandhi, My Father The Wind That Shakes the Barley A Passage to India Guiana 1838 Ramanujan
In [47]:	<pre>import pickle</pre>
In [63]:	<pre>pickle.dump(new,open('movie_list.pkl','wb')) pickle.dump(similarity,open('similarity.pkl','wb'))</pre>