import numpy as np

import pandas as pd

movies=pd.read\_csv('tmdb\_5000\_movies.csv')

credits=pd.read\_csv('tmdb\_5000\_credits.csv')

credits.head()

movies = movies.merge(credits,on='title')

movies.head()

movies = movies[['movie\_id','title','overview','genres','keywords','cast','crew']]

movies.head()

import ast

def convert(text):

L = []

for i in ast.literal\_eval(text):

L.append(i['name'])

return L

movies.dropna(inplace=True)

movies['genres'] = movies['genres'].apply(convert)

movies.head()

movies['keywords'] = movies['keywords'].apply(convert)

def convert3(text):

L = []

counter = 0

for i in ast.literal\_eval(text):

if counter < 3:

L.append(i['name'])

counter+=1

return L

movies.head()

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

def fetch\_director(text):

L = []

for i in ast.literal\_eval(text):

if i['job'] == 'Director':

L.append(i['name'])

return L

movies['crew'] = movies['crew'].apply(fetch\_director)

movies.head()

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

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])

movies.head()

movies['tags']=movies['overview']+movies['genres']+movies['keywords']+movies['cast']+movies['crew']

movies.head()

new\_df=movies[['movie\_id','title','tags']]

new\_df

new\_df['tags']=new\_df['tags'].apply(lambda x:" ".join(x))

new\_df.head()

new\_df['tags'][0]

new\_df['tags']=new\_df['tags'].apply(lambda x:x.lower())

new\_df.head()

!pip install scikit-learn

! pip install nltk

new\_df.head(1)

import nltk

from nltk.stem.porter import PorterStemmer

ps = PorterStemmer()

def stem(text):

y = []

for i in text.split():

y.append(ps.stem(i))

y.append(" ")

return " ".join(y)

from sklearn.feature\_extraction.text import CountVectorizer

cv = CountVectorizer(max\_features=5000,stop\_words='english')

vectors = cv.fit\_transform(new\_df['tags']).toarray()

vectors

(cv.get\_feature\_names())

ps.stem('loved')

new\_df['tags'] = new\_df['tags'].apply(stem)

from sklearn.metrics.pairwise import cosine\_similarity

similarity = cosine\_similarity(vectors)

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

def recommend(movie):

movie\_index = new\_df[new\_df['title']==movie].index[0]

distances = similarity[movie\_index]

movies\_list=sorted(list(enumerate(distances)),reverse = True,key=lambda x:x[1])[1:7]

for i in movies\_list[1:12]:

print(new\_df.iloc[i[0]].title)

recommend('Batman')

new\_df

new\_df.to\_dict()

import pickle

pickle.dump(new\_df,open('movies.pkl','wb'))

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

pickle.dump(new\_df.to\_dict(),open('movie\_dict.pkl','wb'))

recommend('Avatar')