# -\*- coding: utf-8 -\*-

"""Untitled1.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/115e5l-vQjyLKm7nR7UaeBLXGkBxQVOiV

"""

1 import numpy as np

import pandas as pd

import difflib

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

2 # loading the data from the csv file to apandas dataframe

movies\_data = pd.read\_csv('/content/movies.csv')

3 # printing the first 5 rows of the dataframe

movies\_data.head()

4# number of rows and columns in the data frame

movies\_data.shape

5# selecting the relevant features for recommendation

selected\_features = ['genres','keywords','tagline','cast','director']

print(selected\_features)

6# replacing the null valuess with null string

for feature in selected\_features:

movies\_data[feature] = movies\_data[feature].fillna('')

7# combining all the 5 selected features

combined\_features = movies\_data['genres']+' '+movies\_data['keywords']+' '+movies\_data['tagline']+' '+movies\_data['cast']+' '+movies\_data['director']

8print(combined\_features)

9 # converting the text data to feature vectors

vectorizer = TfidfVectorizer()

feature\_vectors = vectorizer.fit\_transform(combined\_features)

10print(feature\_vectors)

11 # getting the similarity scores using cosine similarity

similarity = cosine\_similarity(feature\_vectors)

print(similarity)

print(similarity.shape)

12# getting the movie name from the user

movie\_name = input(' Enter your favourite movie name : ')

list\_of\_all\_titles = movies\_data['title'].tolist()

print(list\_of\_all\_titles)

13 # finding the close match for the movie name given by the user

find\_close\_match = difflib.get\_close\_matches(movie\_name, list\_of\_all\_titles)

print(find\_close\_match)

close\_match = find\_close\_match[0]

print(close\_match)

14# finding the index of the movie with title

index\_of\_the\_movie = movies\_data[movies\_data.title == close\_match]['index'].values[0]

print(index\_of\_the\_movie)

15# getting a list of similar movies

similarity\_score = list(enumerate(similarity[index\_of\_the\_movie]))

print(similarity\_score)

len(similarity\_score)

sorted\_similar\_movies = sorted(similarity\_score, key = lambda x:x[1], reverse = True)

print(sorted\_similar\_movies)

16 # print the name of similar movies based on the index

print('Movies suggested for you : \n')

i = 1

for movie in sorted\_similar\_movies:

index = movie[0]

title\_from\_index = movies\_data[movies\_data.index==index]['title'].values[0]

if (i<10):

print(i, '.',title\_from\_index)

i+=1

17 movie\_name = input(' Enter your favourite movie name : ')

list\_of\_all\_titles = movies\_data['title'].tolist()

find\_close\_match = difflib.get\_close\_matches(movie\_name, list\_of\_all\_titles)

close\_match = find\_close\_match[0]

index\_of\_the\_movie = movies\_data[movies\_data.title == close\_match]['index'].values[0]

similarity\_score = list(enumerate(similarity[index\_of\_the\_movie]))

sorted\_similar\_movies = sorted(similarity\_score, key = lambda x:x[1], reverse = True)

print('Movies suggested for you : \n')

i = 1

for movie in sorted\_similar\_movies:

index = movie[0]

title\_from\_index = movies\_data[movies\_data.index==index]['title'].values[0]

if (i<10):

print(i, '.',title\_from\_index)

i+=1