#!/usr/bin/env python

# coding: utf-8

# In[1]:

import numpy as np

import pandas as pd

import warnings

# In[2]:

warnings.filterwarnings('ignore')

# In[3]:

columns\_name=['user\_id','item\_id','rating','timestamp']

df=pd.read\_csv('u.data',sep="\t",names=columns\_name)

# In[4]:

df.head()

# In[5]:

df.shape

# In[6]:

df['user\_id']

# In[7]:

df['user\_id'].nunique()

# In[8]:

df['item\_id'].nunique()

# In[9]:

movies\_title=pd.read\_csv('u.item',sep="\|",header=None)

# In[10]:

movies\_title.shape

# In[11]:

movies\_titles=movies\_title[[0,1]]

movies\_titles.columns=["item\_id","title"]

movies\_titles.head()

# In[12]:

df=pd.merge(df,movies\_titles,on="item\_id")

# In[13]:

df

# In[14]:

df.tail()

# In[15]:

ratings=pd.DataFrame(df.groupby('title').mean()['rating'])

# In[16]:

ratings.head()

# In[17]:

ratings['num of ratings']=pd.DataFrame(df.groupby('title').count()['rating'])

# #Create the recommendar System

# In[18]:

df.head()

# In[19]:

moviemat=df.pivot\_table(index="user\_id",columns="title",values="rating")

# In[20]:

moviemat.head()

# In[21]:

starwars\_user\_ratings=moviemat['Star Wars (1977)']

# In[22]:

starwars\_user\_ratings.head(20)

# In[23]:

similar\_to\_starwars=moviemat.corrwith(starwars\_user\_ratings)

# In[24]:

similar\_to\_starwars

# In[25]:

corr\_starwars=pd.DataFrame(similar\_to\_starwars,columns=['correlation'])

# In[26]:

corr\_starwars.dropna(inplace=True)

# In[27]:

corr\_starwars

# In[28]:

corr\_starwars.head()

# In[29]:

corr\_starwars.sort\_values('correlation',ascending=False).head(10)

# In[30]:

ratings

# In[31]:

corr\_starwars=corr\_starwars.join(ratings['num of ratings'])

# In[32]:

corr\_starwars

# In[33]:

corr\_starwars.head()

# In[34]:

corr\_starwars[corr\_starwars['num of ratings']>100].sort\_values('correlation',ascending=False)

# In[38]:

def predictMovies(movie):

movie\_user\_ratings=moviemat[movie]

similar\_to\_movie=moviemat.corrwith(movie\_user\_ratings)

corr\_movie=pd.DataFrame(similar\_to\_movie,columns=['correlation'])

corr\_movie.dropna(inplace=True)

corr\_movie=corr\_movie.join(ratings['num of ratings'])

prediction=corr\_movie[corr\_movie['num of ratings']>100].sort\_values('correlation',ascending=False)

return prediction

# In[39]:

my\_movie=predictMovies("12 Angry Men (1957)")

# In[41]:

my\_movie.head(10)

Output: 