Tugas Machine Learning Praktikum

Sistem Rekomendasi



Oleh :

Ferry Triwantono – 082111633094

**FAKULTAS SAINS DAN TEKNOLOGI**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS AIRLANGGA**

**2023**

**Sistem Rekomendasi**

Syntax beserta output :

import pandas as pd

from scipy import sparse

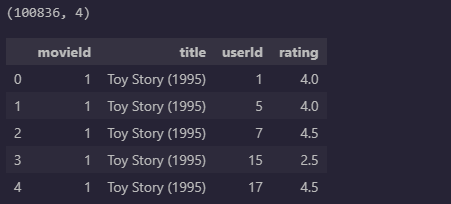
ratings = pd.read\_csv(r'D:\Coolyeah\Mata Kuliah\SMT 4\Machine Learning Praktikum\Modul 8\ratings.csv')

movies = pd.read\_csv(r'D:\Coolyeah\Mata Kuliah\SMT 4\Machine Learning Praktikum\Modul 8\movies.csv')

ratings = pd.merge(movies,ratings).drop(['genres','timestamp'], *axis*=1)

print(ratings.shape)

ratings.head()



Penggabungan dataframe Ratings dengan movies menggunakan fungsi merge() dari library pandas yang memungkinkan menggabungkan dua dataframe berdasarkan kolom yang sama pada kedua data. Kemudian setelah digabungkan, kolom ‘genres’ dan ‘timestamp’ dihapus dari dataframe menggunakan fungsi drop() dari library pandas

Output dari code diatas adalah berupa ukuran dataframe ‘ratings’ setelah dilakukan merge serta penghapusan kolom ‘genres’ dan ‘timestamp dan memunculkan 5 baris pertama dari dataframe ‘ratings’ yang telah diolah

userRatings = ratings.pivot\_table(*index*=['userId'], *columns*=['title'], *values*='rating')

userRatings.head()

print("Before : ", userRatings.shape)

userRatings = userRatings.dropna(*thresh*=10, *axis*=1).fillna(0, *axis*=1)

print("After : ", userRatings.shape)

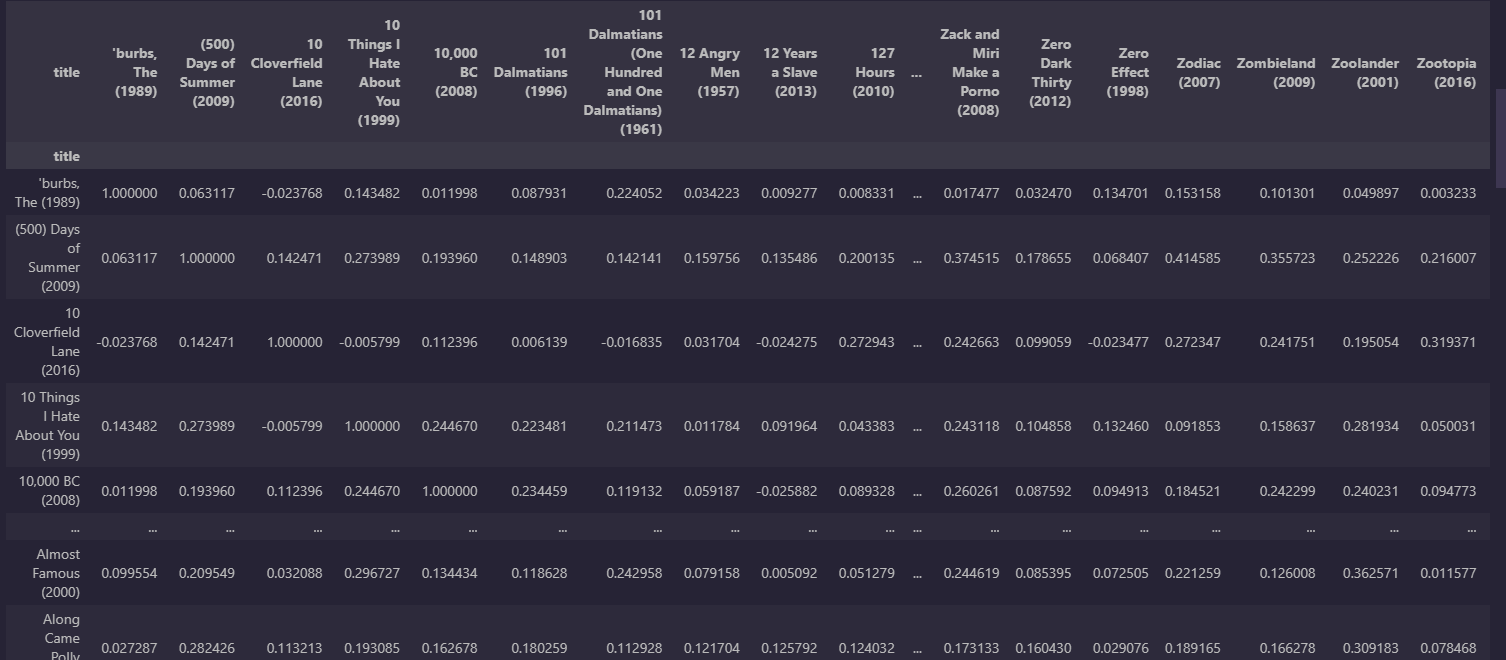


Disini terjadi pengubahan dataframe ‘ratings’ menjadi pivot table yang akan menunjukan nilai rating untuk setiap user pada setiap judul film yang diberikan. Dengan parameter index diisi dengan kolom ‘userId’, parameter columns diisi dengan kolom ‘title’, dan parameter values diisi dengan kolom ‘rating’.

Kemudian ukuran dari pivot table sebelum dan sesudah penghapusan kolom dengan nilai null dicetak ke layer menggunakan fungsi shape(). Kemudian ada syntax dropna() untuk menghapus kolom dan fillna() untuk mengisi kolom yang memiliki nilai null dengan 0

corrMatrix = userRatings.corr(*method*='pearson')

corrMatrix.head(100)



def get\_similar(*movie\_name*,*rating*):

    similar\_ratings = corrMatrix[*movie\_name*]\*(*rating*-2.5)

    similar\_ratings = similar\_ratings.sort\_values(*ascending*=False)

*#print(type(similar\_ratings))*

    return similar\_ratings

Penggunaan fungsi get\_similar yang menggunakan 2 parameter, yaitu ‘movie\_name’ dan ‘rating’ akan menghasilkan daftar film film yang mirip dengan film yang dipilih berdasarkan perhitungan korelasi antara film tersebut

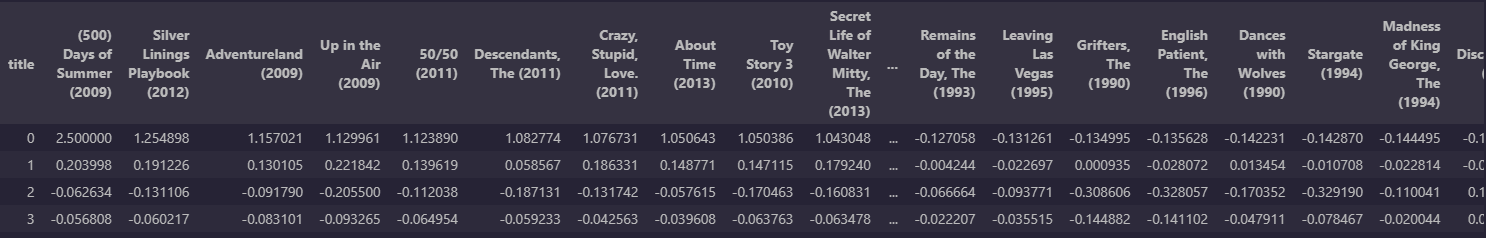
romantic\_lover = [("(500) Days of Summer (2009)",5),("Alice in Wonderland (2010)",3),("Aliens (1986)",1),("2001: A Space Odyssey (1968)",2)]

similar\_movies = pd.DataFrame()

for movie,rating in romantic\_lover:

    similar\_movies = similar\_movies.append(get\_similar(movie,rating),*ignore\_index* = True)

similar\_movies.head(10)



Fungsi get\_similar() digunakan lagi pada setiap film dan rating dalam list ‘romantic\_lover’ untuk mencari daftar film film yang mirip dengan film yang dipilih pada list ‘romantic\_lover’. Hasil dari fungsi tersebut akan di append ke dataframe ‘similar\_movies’

similar\_movies.sum().sort\_values(*ascending*=False).head(20)



Penghitungan nilai kumulatif dari seluruh nilai korelasi pada dataframe ‘similar\_movies’ untuk setiap judul film. Kemudian, hasil perhitungan tersebut akan diurutkan dari nilai yang paling besar ke yang paling kecil.

action\_lover = [("Amazing Spider-Man, The (2012)",5),("Mission: Impossible III (2006)",4),("Toy Story 3 (2010)",2),("2 Fast 2 Furious (Fast and the Furious 2, The) (2003)",4)]

similar\_movies = pd.DataFrame()

for movie,rating in action\_lover:

    similar\_movies = similar\_movies.append(get\_similar(movie,rating),*ignore\_index* = True)

similar\_movies.head(10)

similar\_movies.sum().sort\_values(*ascending*=False).head(20)



Sama halnya dengan ‘romantic\_lover’ namun hanya diganti pada list film apa saja yang dipilih dan dimasukan ke dataframe ‘action lover’