Laporan Sistem Rekomendasi

Pembelajaran Mesin (Praktikum)
SI –I1

187221048 | Muhammad Hanif Abdurrahman S|

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS AIRLANGGA
2024

1

```
#import library
import pandas as pd
from scipy import sparse
```

2

```
# import dataset
ratings = pd.read_csv('ratings.csv')
movies = pd.read_csv('movies.csv')
```

3

```
#3 baca dataset ratings
print("Data Ratings")
print(ratings)
```

Output:

```
Data Ratings
        userId  movieId  rating   timestamp
0            1        1     4.0   964982703
1            1        3     4.0   964981247
2            1        6     4.0   964982224
3            1       47     5.0   964983815
4            1       50     5.0   964982931
...        ...      ...     ...         ...
100831     610   166534     4.0  1493848402
100832     610   168248     5.0  1493850091
100833     610   168250     5.0  1494273047
100834     610   168252     5.0  1493846352
100835     610   170875     3.0  1493846415

[100836 rows x 4 columns]
```

4

```
#4 cek null data, hapus apabila ada data yang kosong
missing_values = ratings.isnull().sum()
print("\nJumlah missing value untuk setiap atribut:\n", missing_values)
```

Output:

```
Jumlah missing value untuk setiap atribut:
 userId       0
movieId      0
rating       0
timestamp    0
dtype: int64
```

5

```
#5 baca dataset movies
print("\nData Movies")
print(movies)
```

Output:

	Data Movies movieId title genres 0 1 Toy Story (1995) Adventure|Animation|Children|Comedy|Fantasy 1 2 Jumanji (1995) Adventure|Children|Fantasy 2 3 Grumpier Old Men (1995) Comedy|Romance 3 4 Waiting to Exhale (1995) Comedy|Drama|Romance 4 5 Father of the Bride Part II (1995) Comedy ... ... 9737 193581 Black Butler: Book of the Atlantic (2017) Action|Animation|Comedy|Fantasy 9738 193583 No Game No Life: Zero (2017) Animation|Comedy|Fantasy 9739 193585 Flint (2017) Drama 9740 193587 Bungo Stray Dogs: Dead Apple (2018) Action|Animation 9741 193609 Andrew Dice Clay: Dice Rules (1991) Comedy [9742 rows x 3 columns]

6	`#6 cek null data, hapus apabila ada data yang kosong`
	`missing_values = movies.isnull().sum()`
	`print("\nJumlah missing value untuk setiap atribut:\n", missing_values)`
	Output:
	Jumlah missing value untuk setiap atribut:
	movieId 0
	title 0
	genres 0
	dtype: int64

7	`ratings.shape` **ratings.shape adalah sebuah properti dari DataFrame pada Pandas yang mengembalikan tupel berisi jumlah baris dan kolom dalam DataFrame tersebut. Secara khusus, ratings.shape[0] akan memberikan jumlah baris (jumlah entri atau data), sedangkan ratings.shape[1] akan memberikan jumlah kolom (jumlah atribut atau fitur). Dengan menggunakan ratings.shape, Anda dapat dengan cepat mendapatkan gambaran tentang ukuran data yang sedang Anda kerjakan.** **Output:** **(100836, 4)** **Dalam output tersebut, 100836 merupakan jumlah baris dalam data sedangkan 4 merupakan jumlah kolom data ratings.**
8	`movies.shape` ***# jelaskan maksud statement ini, dan jelaskan hasilnya*** **movies.shape adalah sebuah properti dari DataFrame pada Pandas yang mengembalikan tupel berisi jumlah baris dan kolom dalam DataFrame tersebut. Secara khusus, movies.shape[0] akan memberikan jumlah baris (jumlah entri atau data), sedangkan movies.shape[1] akan memberikan jumlah kolom (jumlah atribut atau fitur). Dengan menggunakan movies.shape, Anda dapat dengan cepat mendapatkan gambaran tentang ukuran data yang sedang Anda kerjakan.** **Output:** **(9742, 3)**

	Dalam output tersebut, 9742 merupakan jumlah baris dalam data sedangkan 3 merupakan jumlah kolom data ratings.

9	`#9 sebutkan atribut-atribut yang sama antara dataset ratings dan movies`
	`# Menampilkan nama kolom dari dataset ratings`
	`print("\nKolom dataset ratings:")`
	`print(ratings.columns)`
	
	`# Menampilkan nama kolom dari dataset movies`
	`print("\nKolom dataset movies:")`
	`print(movies.columns)`
	Output:
	`Kolom dataset ratings:`
	`Index(['userId', 'movieId', 'rating', 'timestamp'], dtype='object')`
	
	`Kolom dataset movies:`
	`Index(['movieId', 'title', 'genres'], dtype='object')`
	Kedua data tersebut memiliki nama kolom yang sama yakni pada movieId

10	`# melakukan merge dataset ratings dan movies`
	
	`ratings = pd.merge(movies,ratings).drop(['genres','timestamp'],axis=1)`
	Ini menggabungkan kedua dataset berdasarkan kolom 'title', lalu menghapus kolom 'genres' dan 'timestamp'.
	`print(ratings.shape)`
	
	`userRatings = ratings.pivot_table(index=['userId'],columns=['title'],values='rating')`
	ratings.shape kemudian mencetak ukuran DataFrame baru setelah penggabungan.userRatings= ratings.pivot_table(index=['userId'],columns=['title'],values='rating') kemudian membuat pivot table dari DataFrame ratings yang baru saja dibuat, dengan userId sebagai indeks, judul film sebagai kolom, dan nilai rating sebagai nilai.
	Output:

```
Rating Film
title   '71 (2014)  ...  À nous la liberté (Freedom for Us) (1931)
userId              ...
1              NaN  ...                                        NaN
2              NaN  ...                                        NaN
3              NaN  ...                                        NaN
4              NaN  ...                                        NaN
5              NaN  ...                                        NaN
...            ...  ...                                        ...
606            NaN  ...                                        NaN
607            NaN  ...                                        NaN
608            NaN  ...                                        NaN
609            NaN  ...                                        NaN
610            4.0  ...                                        NaN

[610 rows x 9719 columns]
```

userRatings

Penggabungan kedua data tersebut menghasilkan table dari DataFrame ratings yang baru saja dibuat, dengan userId sebagai indeks, judul film sebagai kolom, dan nilai rating sebagai nilai.

11

```
# amati bentuk data
print("Before: ",userRatings.shape)
userRatings = userRatings.dropna(thresh=10, axis=1).fillna(0,axis=1)
print("After: ",userRatings.shape)
```

Output:

Before: (610, 9719)

After: (610, 2269)

Ini menghapus film yang memiliki kurang dari 10 nilai rating non-null, dan mengganti nilai null dengan 0.

12

```
#menghitung matriks korelasi
corrMatrix = userRatings.corr(method='pearson')
corrMatrix
```

Output:

```
Matriks Korelasi
title                            'burbs, The (1989)  (500) Days of Summer (2009)  ...  xXx (2002)  ¡Three Amigos! (
1986)
title                                                                            ...
'burbs, The (1989)                         1.000000                    0.063117  ...    0.062174              0.3
53194
(500) Days of Summer (2009)                0.063117                    1.000000  ...    0.241092              0.1
25905
10 Cloverfield Lane (2016)                -0.023768                    0.142471  ...    0.096638              0.0
02733
10 Things I Hate About You (1999)          0.143482                    0.273989  ...    0.130813              0.1
10612
10,000 BC (2008)                           0.011998                    0.193960  ...    0.203002              0.0
83518
...                                             ...                         ...  ...         ...
 ...
Zoolander (2001)                           0.049897                    0.252226  ...    0.338034              0.1
09455
Zootopia (2016)                            0.003233                    0.216007  ...    0.200762              0.0
20595
eXistenZ (1999)                            0.187953                    0.053614  ...    0.163022              0.1
38611
xXx (2002)                                 0.062174                    0.241092  ...    1.000000              0.0
65673
¡Three Amigos! (1986)                      0.353194                    0.125905  ...    0.065673              1.0
00000

[2269 rows x 2269 columns]
```

13

```
#romantic_lover movie list
romantic_lover = [
    ("(500) Days of Summer (2009)", 5),
    ("Alice in Wonderland (2010)", 3),
    ("Aliens (1986)", 1),
    ("2001: A Space Odyssey (1968)", 2)
]
```

14

```
# find similar movie with romantic lover list

def get_similar(movie_name, rating):
    similar_ratings = corrMatrix[movie_name] * (rating)
    similar_ratings = similar_ratings.sort_values(ascending=False)
    return similar_ratings

similar_movies = pd.concat([get_similar(movie, rating) for movie, rating in
romantic_lover], axis=1)
similar_movies.head(10)
```

Output:

	(500) Days of Summer (2009) ... 2001: A Space Odyssey (1968) title ... (500) Days of Summer (2009) 5.000000 ... 0.227232 Silver Linings Playbook (2012) 2.509796 ... 0.240869 Adventureland (2009) 2.314041 ... 0.332404 Up in the Air (2009) 2.259921 ... 0.373062 50/50 (2011) 2.247781 ... 0.259818 Descendants, The (2011) 2.165548 ... 0.236932 Crazy, Stupid, Love. (2011) 2.153462 ... 0.170254 About Time (2013) 2.101287 ... 0.158432 Toy Story 3 (2010) 2.100772 ... 0.255052 Secret Life of Walter Mitty, The (2013) 2.086095 ... 0.253914 [10 rows x 4 columns] **Fungsi get_similar menghitung peringkat kesamaan antara setiap film dalam daftar film romantis dan semua film lainnya berdasarkan matriks korelasi. Hasilnya adalah daftar film-film yang paling mirip dengan film-film romantis yang disukai pengguna, beserta peringkat kesamaannya.**

15	`comedy = [`
	`("Grumpier Old Men (1995)", 5),`
	`("Toy Story (1995)", 3),`
	`("Sabrina (1995)", 1),`
	`("Four Rooms (1995)", 2)`
	`]`

16	`similar_movies = pd.concat([get_similar(movie, rating) for movie, rating in comedy], axis=1)`
	`display(similar_movies.head(10))`
	Output:
	Grumpier Old Men (1995) Toy Story (1995) Sabrina (1995) Four Rooms (1995)
	title
	Grumpier Old Men (1995) 5.000000 0.519638 0.351513 0.319497
	Grumpy Old Men (1993) 2.077235 0.429895 0.081270 0.421006
	Striptease (1996) 2.040742 0.474861 0.275325 0.454182
	Nutty Professor, The (1996) 1.942527 1.050884 0.251046 0.161868
	Son in Law (1993) 1.893008 0.376102 0.174058 0.077019
	Bio-Dome (1996) 1.859691 0.526940 0.214180 0.296249
	Father of the Bride Part II (1995) 1.850854 0.577422 0.429890 0.380823
	Twister (1996) 1.826910 0.876923 0.282102 0.218861
	Juror, The (1996) 1.810676 0.261867 0.320087 0.159238
	Beverly Hillbillies, The (1993) 1.777787 0.401713 0.186673 0.009358
	Fungsi get_similar menghitung peringkat kesamaan antara setiap film dalam daftar film komedi dan semua film lainnya berdasarkan matriks korelasi. Hasilnya adalah daftar film-film yang paling mirip dengan film-film komedi yang disukai pengguna, beserta peringkat kesamaannya.