Yangzhou University

Software Data Analysis 2022

鲁卡娅+ MH20055

Recommender System (content based filtering)

dataset resource : https://grouplens.org/datasets/movielens/100k/

In [327…
```
import numpy as np
import pandas as pd
```

In [328…
```
#### dataFrames
unames = ['user_id','age','gender','occupation','zipcode']
users = pd.read_csv('ml-100k/u.user',sep='|',names=unames)
rnames=['user_id','item id','rating','timestamp']
ratings = pd.read_csv('ml-100k/u.data',sep='\t',names=rnames)
mnames=['item id','title','realease_date','video release date','IMDb URL','unknown','Action','Adventure','Animation','Children',
        'Comedy','Crime','Documentary','Drama','Fantasy',
               'Film-Noir','Horror', 'Musical','Mystery','Romance','Sci-Fi',
               'Thriller','War','Western']
movie_df = pd.read_csv('ml-100k/u.item',sep='|',names=mnames,encoding='latin-1')
```

In [ ]:

In [ ]:

In [329…
```
user_df = users.loc[:,['user_id','gender']]
ratings_df=ratings.loc[:,['user_id','rating']]
```

In [330…
```
rating_df=ratings_df.merge(user_df)
```

In [331…
```
rating_df
```

Out[331…

	user_id	rating	gender
0	196	3	M
1	196	4	M
2	196	4	M
3	196	3	M
4	196	5	M
...	...	...	...
99995	941	5	M
99996	941	3	M
99997	941	5	M
99998	941	4	M
99999	941	4	M

100000 rows × 3 columns

In [332…
```
rating_df.groupby('gender').rating.std()
```

Out[332…
```
gender
F    1.170951
M    1.109556
Name: rating, dtype: float64
```

In [333…
```
rating_df.groupby('gender').rating.apply(pd.Series.std)
```

Out[333…
```
gender
F    1.170951
M    1.109556
Name: rating, dtype: float64
```

In [334…
```
df1=rating_df.groupby(['user_id','gender']).apply(np.mean)
```

```
C:\Users\Rukaia\AppData\Local\Programs\Python\Python39\lib\site-packages\numpy\core\fromnumeric.py:3438: FutureWarning: Dropping of nuisance columns in DataFr
ame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reductio
n.
  return mean(axis=axis, dtype=dtype, out=out, **kwargs)
```

In [335…
```
df1.groupby('gender').rating.std()
```

Out[335…
```
gender
F    0.481241
M    0.430076
Name: rating, dtype: float64
```

In [336…
```
pd.pivot_table(df1, values = 'rating', index = 'gender', aggfunc = pd.Series.std)
```

Out[336…

	rating
gender	
F	0.481241
M	0.430076

In [337…
```
pd.pivot_table(rating_df, index = ['user_id','gender'], values = 'rating')
```

Out[337…

		rating
user_id	gender	
1	M	3.610294
2	F	3.709677
3	M	2.796296
4	M	4.333333
5	F	2.874286
...	...	...
939	F	4.265306
940	M	3.457944
941	M	4.045455
942	F	4.265823
943	M	3.410714

943 rows × 1 columns

In [338…
```
t = pd.pivot_table(rating_df, index = ['user_id','gender'], values = 'rating')
female = t.query("gender == ['F']")
pd.Series.std(female)
```

Out[338…
```
rating    0.481241
dtype: float64
```

In [339…
```
t = pd.pivot_table(rating_df, index = ['user_id','gender'], values = 'rating')
male = t.query("gender == ['M']")
pd.Series.std(male)
```

Out[339…
```
rating    0.430076
dtype: float64
```

Movei Recommandation

In [340…
```
#### slicing dframe is like if index position or refereance is involved iloc should be used or if exact column name then loc can be used
## df_movie=df_movie.iloc[:,1:2]
## or
movie_df=movie_df.loc[:,["item id","title"]]
movie_df
```

Out[340…

	item id	title
0	1	Toy Story (1995)
1	2	GoldenEye (1995)
2	3	Four Rooms (1995)
3	4	Get Shorty (1995)
4	5	Copycat (1995)
...	...	...
1677	1678	Mat' i syn (1997)
1678	1679	B. Monkey (1998)
1679	1680	Sliding Doors (1998)
1680	1681	You So Crazy (1994)
1681	1682	Scream of Stone (Schrei aus Stein) (1991)

1682 rows × 2 columns

In [341…
```
#### now merge this with ratings
ratings
```

Out[341…

	user_id	item id	rating	timestamp
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596
...	...	...	...	...
99995	880	476	3	880175444
99996	716	204	5	879795543
99997	276	1090	1	874795795
99998	13	225	2	882399156
99999	12	203	3	879959583

100000 rows × 4 columns

In [342…
```
final_df=pd.merge(ratings,movie_df,on="item id")
### the on column should be named same like i named item id on ratings and 'id' on movie_df ,got error
```

In [343…
```
final_df
```

Out[343…

	user_id	item id	rating	timestamp	title
0	196	242	3	881250949	Kolya (1996)
1	63	242	3	875747190	Kolya (1996)
2	226	242	5	883888671	Kolya (1996)
3	154	242	3	879138235	Kolya (1996)
4	306	242	5	876503793	Kolya (1996)
...	...	...	...	...	...
99995	840	1674	4	891211682	Mamma Roma (1962)
99996	655	1640	3	888474646	Eighth Day, The (1996)
99997	655	1637	3	888984255	Girls Town (1996)
99998	655	1630	3	887428735	Silence of the Palace, The (Saimt el Qusur) (1...
99999	655	1641	3	887427810	Dadetown (1995)

100000 rows × 5 columns

In [344…
```
final_df.drop("timestamp",axis=1,inplace=True)
```

In [345…
```
#### 1.find averege ratings for movies
#### 2.the num of ratings for each movie.
#### 3.and finally make a ready dataframe with the avg rating and counts
```

In [346…
```
final_df.groupby("title")["rating"].mean().sort_values(ascending=False)
```

Out[346…
```
title
They Made Me a Criminal (1939)                 5.0
Marlene Dietrich: Shadow and Light (1996)      5.0
Saint of Fort Washington, The (1993)           5.0
Someone Else's America (1995)                  5.0
Star Kid (1997)                                5.0
                                              ...
Eye of Vichy, The (Oeil de Vichy, L') (1993)   1.0
King of New York (1990)                        1.0
Touki Bouki (Journey of the Hyena) (1973)      1.0
Bloody Child, The (1996)                       1.0
Crude Oasis, The (1995)                        1.0
Name: rating, Length: 1664, dtype: float64
```

In [347…
```
count_ratings=final_df.groupby("title")["rating"].count().sort_values(ascending=False)
```

In [348…
```
movie_details = pd.DataFrame(final_df.groupby("title")["rating"].mean().sort_values(ascending=False))
```

In [349…
```
movie_details["count_ratings"]= final_df.groupby("title")["rating"].count()
```

In [350…
```
movie_details
```

Out[350…

	rating	count_ratings
title		
They Made Me a Criminal (1939)	5.0	1
Marlene Dietrich: Shadow and Light (1996)	5.0	1
Saint of Fort Washington, The (1993)	5.0	2
Someone Else's America (1995)	5.0	1
Star Kid (1997)	5.0	3
...	...	...
Eye of Vichy, The (Oeil de Vichy, L') (1993)	1.0	1
King of New York (1990)	1.0	1
Touki Bouki (Journey of the Hyena) (1973)	1.0	1
Bloody Child, The (1996)	1.0	1
Crude Oasis, The (1995)	1.0	1

1664 rows × 2 columns

In [351…
```
movie_details.sort_values(["rating","count_ratings"],ascending=False).head(20)
#### 5 and 1  rated movies got few ratings
```

Out[351…

	rating	count_ratings
title		
Star Kid (1997)	5.000000	3
Prefontaine (1997)	5.000000	3
Saint of Fort Washington, The (1993)	5.000000	2
Santa with Muscles (1996)	5.000000	2
They Made Me a Criminal (1939)	5.000000	1
Marlene Dietrich: Shadow and Light (1996)	5.000000	1
Someone Else's America (1995)	5.000000	1
Great Day in Harlem, A (1994)	5.000000	1
Aiqing wansui (1994)	5.000000	1
Entertaining Angels: The Dorothy Day Story (1996)	5.000000	1
Pather Panchali (1955)	4.625000	8
Maya Lin: A Strong Clear Vision (1994)	4.500000	4
Some Mother's Son (1996)	4.500000	2
Anna (1996)	4.500000	2
Everest (1998)	4.500000	2
Close Shave, A (1995)	4.491071	112
Schindler's List (1993)	4.466443	298
Wrong Trousers, The (1993)	4.466102	118
Casablanca (1942)	4.456790	243
Wallace & Gromit: The Best of Aardman Animation (1996)	4.447761	67

In [352…
```
##### for recommandatin , we guess a user watched and rated a movie named "Toy Story"
#### 1.we need to make a pivot table for users and rated movie with the use of dataFrame "final_df"
#### 2.we need to make a correlation with the pivot table named as "moviemat" and a selected movie with the use of dataFrame "movie_details"
```

In [353…
```
moviemat=pd.pivot_table(data=final_df,index="user_id",columns="title",values="rating")
```

In [354…
```
moviemat
```

Out[354…

title	'Til There Was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	39 Steps, The (1935)	...	Yankee Zulu (1994)	Year of the Horse (1997)	You So Crazy (1994)	Young Frankenstein (1974)	Young Guns (1988)	Young Guns II (1990)	Young Poisoner's Handbook, The (1995)	Zeus and Roxanne (1997)	unknown	Á köldum klaka (Cold Fever) (1994)
user_id																					
1	NaN	NaN	2.0	5.0	NaN	NaN	3.0	4.0	NaN	NaN	...	NaN	NaN	NaN	5.0	3.0	NaN	NaN	NaN	4.0	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	2.0	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	2.0	NaN	NaN	NaN	NaN	4.0	NaN	NaN	...	NaN	NaN	NaN	4.0	NaN	NaN	NaN	NaN	4.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
939	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
940	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
941	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
942	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.0	NaN	3.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
943	NaN	NaN	NaN	NaN	NaN	2.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	4.0	3.0	NaN	NaN	NaN	NaN

943 rows × 1664 columns

In [355…
```
ToyStory_userRatings= moviemat["Toy Story (1995)"]
ToyStory_userRatings
```

Out[355…
```
user_id
1      5.0
2      4.0
3      NaN
4      NaN
5      4.0
      ...
939    NaN
940    NaN
941    5.0
942    NaN
943    NaN
Name: Toy Story (1995), Length: 943, dtype: float64
```

In [356…
```
similar_to_ToyStory = moviemat.corrwith(ToyStory_userRatings)
```

```
C:\Users\Rukaia\AppData\Local\Programs\Python\Python39\lib\site-packages\numpy\lib\function_base.py:2683: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
C:\Users\Rukaia\AppData\Local\Programs\Python\Python39\lib\site-packages\numpy\lib\function_base.py:2542: RuntimeWarning: divide by zero encountered in true_d
ivide
  c *= np.true_divide(1, fact)
```

In [357…
```
similar_to_ToyStory
```

Out[357…
```
title
'Til There Was You (1997)                0.534522
1-900 (1994)                                  NaN
101 Dalmatians (1996)                    0.232118
12 Angry Men (1957)                      0.334943
187 (1997)                               0.651857
                                           ...
Young Guns II (1990)                     0.146312
Young Poisoner's Handbook, The (1995)   -0.026402
Zeus and Roxanne (1997)                  0.447914
unknown                                  0.440959
Á köldum klaka (Cold Fever) (1994)            NaN
Length: 1664, dtype: float64
```

In [358…
```
df_similar_to_toystory = pd.DataFrame(similar_to_ToyStory,columns=["correlation"])
```

In [359…
```
df_similar_to_toystory.dropna(inplace=True)
```

In [360…
```
df_similar_to_toystory
```

Out[360…

	correlation
title	
'Til There Was You (1997)	0.534522
101 Dalmatians (1996)	0.232118
12 Angry Men (1957)	0.334943
187 (1997)	0.651857
2 Days in the Valley (1996)	0.162728
...	...
Young Guns (1988)	0.373463
Young Guns II (1990)	0.146312
Young Poisoner's Handbook, The (1995)	-0.026402
Zeus and Roxanne (1997)	0.447914
unknown	0.440959

1370 rows × 1 columns

In [361…
```
#### now find out the highly correlated values
```

In [362…
```
df_similar_to_toystory
```

Out[362…

	correlation
title	
'Til There Was You (1997)	0.534522
101 Dalmatians (1996)	0.232118
12 Angry Men (1957)	0.334943
187 (1997)	0.651857
2 Days in the Valley (1996)	0.162728
...	...
Young Guns (1988)	0.373463
Young Guns II (1990)	0.146312
Young Poisoner's Handbook, The (1995)	-0.026402
Zeus and Roxanne (1997)	0.447914
unknown	0.440959

1370 rows × 1 columns

In [363…
```
df_similar_to_toystory.sort_values("correlation",ascending=False)
```

Out[363…

	correlation
title	
Old Lady Who Walked in the Sea, The (Vieille qui marchait dans la mer, La) (1991)	1.0
Reckless (1995)	1.0
Stranger, The (1994)	1.0
Ladybird Ladybird (1994)	1.0
Infinity (1996)	1.0
...	...
Slingshot, The (1993)	-1.0
Heavy (1995)	-1.0
Feast of July (1995)	-1.0
Stalker (1979)	-1.0
Love and Death on Long Island (1997)	-1.0

1370 rows × 1 columns

In [364…
```
#### there are a lots of  highly correlatead movies. so we have to filter them more to make the recommandation more useful.
#### we are gonna do it by rating counts. who got rated more
### so combine the 'df_similar_to_toystory' dataframe with 'movie_details' dataframe in order to get the numbers of ratings
```

In [365…
```
movie_details
```

Out[365…

	rating	count_ratings
title		
They Made Me a Criminal (1939)	5.0	1
Marlene Dietrich: Shadow and Light (1996)	5.0	1
Saint of Fort Washington, The (1993)	5.0	2
Someone Else's America (1995)	5.0	1
Star Kid (1997)	5.0	3
...	...	...
Eye of Vichy, The (Oeil de Vichy, L') (1993)	1.0	1
King of New York (1990)	1.0	1
Touki Bouki (Journey of the Hyena) (1973)	1.0	1
Bloody Child, The (1996)	1.0	1
Crude Oasis, The (1995)	1.0	1

1664 rows × 2 columns

In [366…
```
movie_details.loc["101 Dalmatians (1996)",:]
### iloc used for only integer values where loc used for strings too
```

Out[366…
```
rating             2.908257
count_ratings    109.000000
Name: 101 Dalmatians (1996), dtype: float64
```

In [367…
```
df_similar_to_toystory=df_similar_to_toystory.join(movie_details["count_ratings"])
```

In [368…
```
df_similar_to_toystory
```

Out[368…

	correlation	count_ratings
title		
'Til There Was You (1997)	0.534522	9
101 Dalmatians (1996)	0.232118	109
12 Angry Men (1957)	0.334943	125
187 (1997)	0.651857	41
2 Days in the Valley (1996)	0.162728	93
...	...	...
Young Guns (1988)	0.373463	101
Young Guns II (1990)	0.146312	44
Young Poisoner's Handbook, The (1995)	-0.026402	41
Zeus and Roxanne (1997)	0.447914	6
unknown	0.440959	9

1370 rows × 2 columns

In [369…
```
df_similar_to_toystory=df_similar_to_toystory[df_similar_to_toystory["count_ratings"]>100].sort_values('correlation',ascending=False)
```

In [370…
```
df_similar_to_toystory
```

Out[370…

	correlation	count_ratings
title		
Toy Story (1995)	1.000000	452
Craft, The (1996)	0.549100	104
Down Periscope (1996)	0.457995	101
Miracle on 34th Street (1994)	0.456291	101
G.I. Jane (1997)	0.454756	175
...	...	...
Raging Bull (1980)	-0.083523	116
Boogie Nights (1997)	-0.086194	189
Clockwork Orange, A (1971)	-0.087710	221
Apt Pupil (1998)	-0.104066	160
Harold and Maude (1971)	-0.181697	121

334 rows × 2 columns

The end :)

In [ ]: