DM end1 problem 5

Ans.1

N=3
M=5

Import libraries

```
In [11]: import numpy as np
         import pandas as pd
```

Parameters

```
In [12]: min_common_items = 3
```

```
In [13]: def predict_scores(df_sim, ser_target):
             ret = {}
             for item1 in df_sim.index:  # not yet rated by the target user
                 v1 = df_sim.loc[item1]

                 if v1.notnull().sum() < min_common_items: continue
                 v11 = v1[ v1.notnull() ]
                 t11 = ser_target[ v1.notnull() ]
                 pred1 = (v11 * t11).sum() / np.abs(v11).sum()

                 ret[item1] = pred1

             ser_ret = pd.Series(ret)

             return ser_ret.sort_values(ascending=False)
```

Function for user-based collaborative filtering.

arguments: dictionary of scores for the target user
and the number of items to recommend.

ex) get_recomm_by_user_sim({'maguro':1, 'ika':1, 'uni':3, 'awabi':4, 'hirame':4, 'aoyagi':4})
-> return list such as [('akagai', 2.9835603009918303), ('mirugai', 2.945676429588114), ...]

```
In [14]: def get_recomm_by_item_sim(df, target_dic):
             ser_target = pd.Series(target_dic)

             df_scores = df[ ser_target.index ]

             df_scores = df_scores.drop(index=ser_target.index)

             recomm = predict_scores(df_scores, ser_target)

             return recomm
```

```
In [15]: df = pd.read_csv('dm-end1-5.csv', delimiter=',', skiprows=0, header=
         0)
         df.index = df.columns
         print(df.shape)
         print(df.info())
         display(df.head())
```

```
(10, 10)
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, A to J
Data columns (total 10 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   A       10 non-null     float64
 1   B       10 non-null     float64
 2   C       10 non-null     float64
 3   D       10 non-null     float64
 4   E       10 non-null     float64
 5   F       10 non-null     float64
 6   G       10 non-null     float64
 7   H       10 non-null     float64
 8   I       10 non-null     float64
 9   J       10 non-null     float64
dtypes: float64(10)
memory usage: 1.2+ KB
None
```

	A	B	C	D	E	F	G	H	
A	1.000000	-0.130445	0.030770	-0.028525	0.163248	0.048338	0.016468	-0.075529	-0
B	-0.130445	1.000000	0.049838	-0.079938	-0.228926	-0.120330	-0.192542	-0.116864	-0
C	0.030770	0.049838	1.000000	-0.034552	-0.005082	0.096143	0.119985	0.018584	-0
D	-0.028525	-0.079938	-0.034552	1.000000	-0.118688	-0.132196	0.038753	0.070684	-0
E	0.163248	-0.228926	-0.005082	-0.118688	1.000000	-0.067251	0.083375	0.008368	-0

Do recommendation

```
In [16]: recomm = get_recomm_by_item_sim(df, {'A':5, 'B':3, 'C':1,} )
         print('Number of items calculated:', len(recomm))
         print('Recommendation:')
         print(recomm.head())
```

```
Number of items calculated: 7
Recommendation:
E    0.313098
F   -0.087449
G   -1.140757
D   -2.915722
H   -3.363650
dtype: float64
```

Ans.1

E

0.31

```
In [17]: recomm = get_recomm_by_item_sim((df+1)/2, {'D':1, 'E':3, 'F':5,} )
         print('Number of items calculated:', len(recomm))
         print('Recommendation:')
         print(recomm.head())
```

```
Number of items calculated: 7
Recommendation:
C    3.085519
A    3.048295
I    2.998008
G    2.985119
J    2.968894
dtype: float64
```

Ans.2

C

3.09

```
In [ ]:
```