

## ▼ Setting up

```
import numpy as np
import pandas as pd

#Defining variables
target_user = 610
target_movie = 589
N = 2

#"importing" data
data = {
    610: [np.NaN, 0.928571, 0.428571, 0.428571],
    247: [-0.6, 0.600000, -0.100000, -0.100000],
    580: [np.NaN, 0.666667, 0.166667, 0.166667],
    304: [0.4, -0.600000, -0.600000, 0.400000],
    391: [-0.4, 0.600000, -1.400000, 0.600000]
}

#preprocessing
df = pd.DataFrame(data).T
df = df.rename(columns={0: 589, 1: 2571, 2: 593, 3: 110})
df = df.fillna(0)
df.index.name = "userID"
df.columns.name = "movie"

#displaying Data
df
```

	movie	589	2571	593	110
userID					
610		0.0	0.928571	0.428571	0.428571
247		-0.6	0.600000	-0.100000	-0.100000
580		0.0	0.666667	0.166667	0.166667
304		0.4	-0.600000	-0.600000	0.400000
391		-0.4	0.600000	-1.400000	0.600000

## ▼ Preforming User-User filtering

```
user_at = at.drop(target_user)
#grabbing similiar users
user_sims = user_df[user_df.loc[:, target_movie] != 0].apply(lambda y: (y.values*x.values).
#grabbing top 2 users
sorted_user_sims = user_sims.sort_values()[::-1]
N_user_sims = sorted_user_sims.iloc[:N]

#looking up their rating of the target movie and averaging
user_ids = N_user_sims.index
user_ans = (df.loc[user_ids, target_movie].mean())

#displaying ans
user_ans

-0.5
```

Our prediction for user-user filtering is -0.5

## Preforming Item-Item Filtering

```
#going to need to invert the DF
item_df = df.T
x = item_df.loc[target_movie]
item_df = item_df.drop(target_movie)

#grabbing similar movies to the item
movie_sims = item_df.apply(lambda y: (y.values*x.values).sum()/(np.sqrt((y**2).sum())*np.:
sorted_movie_sims = movie_sims.sort_values()[::-1]
N_movie_sims = sorted_movie_sims.iloc[:N]

#figuring out the average rating of movies similar to this movie
movie_ids = N_movie_sims.index
ans = item_df.loc[movie_ids].stack().mean()

#displaying ans
ans

-0.0009523999999999866
```

Our prediction for item-item filtering is -0.0009523999999999866

[Colab paid products](#) - [Cancel contracts here](#)