

추천 시스템을 시작하기 전에

	피자	치킨	김밥	탕수육
고객1	좋다	좋다	X	좋다
고객2	X	X	좋다	X
고객3	좋다	좋다	X	?

고객3에 새로운 음식을 추천해야 함.

In [11]:

```
import numpy as np

user1 = np.array([2,2,1,2])
user2 = np.array([1,1,2,1])
user3 = np.array([2,2,1,0])

rMatrix = np.vstack( (user1, user2, user3))
print(rMatrix.shape)
rMatrix
```

(3, 4)

Out[11]:

```
array([[2, 2, 1, 2],
       [1, 1, 2, 1],
       [2, 2, 1, 0]])
```

사이킷 런의 cosine_similarity함수를 이용한 코사인 유사도

In [12]:

```
from sklearn.metrics.pairwise import cosine_similarity

cosineSim = cosine_similarity(rMatrix)
print(cosineSim.shape)
cosineSim
```

(3, 3)

Out[12]:

```
array([[1.          , 0.83862787, 0.83205029],
       [0.83862787, 1.          , 0.75592895],
       [0.83205029, 0.75592895, 1.          ]])
```

u1	u1		u1	u2		u1	u3
u2	u1		u2	u2		u2	u3
u3	u1		u3	u2		u3	u3

아이템 기반 협업 필터링의 경우

	user1	user2	user3
피자			
치킨			
김밥			
탕수육			

In [13]:

```
rMatrix_t = np.transpose(rMatrix)
print(rMatrix_t.shape)
rMatrix_t
```

(4, 3)

Out[13]:

```
array([[2, 1, 2],
       [2, 1, 2],
       [1, 2, 1],
       [2, 1, 0]])
```

In [14]:

```
cosineSim_t = cosine_similarity(rMatrix_t)
cosineSim_t
```

Out[14]:

```
array([[1.          , 1.          , 0.81649658, 0.74535599],
       [1.          , 1.          , 0.81649658, 0.74535599],
       [0.81649658, 0.81649658, 1.          , 0.73029674],
       [0.74535599, 0.74535599, 0.73029674, 1.          ]])
```

피자와 피자		피자와 치킨		피자와 김밥		피자와 탕수육
치킨과 피자		치킨과 치킨		치킨과 김밥		치킨과 탕수육
김밥과 피자		김밥과 치킨		김밥과 김밥		김밥과 탕수육
탕수육과 피자		탕수육과 치킨		탕수육과 김밥		탕수육과 탕수육

- 탕수육과 피자(or 치킨)은 비슷한 선호도(0.7453559)를 갖는다.

	피자	치킨	김밥	탕수육
고객1	좋다	좋다	X	좋다
고객2	X	X	좋다	X
고객3	좋다	좋다	X	?

모델 기반 협업 필터링

관리자 권한으로 실행

```
pip install scikit-surprise
```

or

```
conda install -c conda-forge scikit-surprise
```

데이터 가져오기

In [34]:

```
from surprise import Dataset
import pandas as pd
```

In [35]:

```
#data = Dataset.load_builtin('ml-100k')
```

SVD 행렬 분해 기법을 이용하여 추천 예측

In [51]:

```
from surprise import SVD
from surprise import Dataset
from surprise import accuracy
from surprise.model_selection import train_test_split
```

In [52]:

```
df = pd.read_csv('../data/grouplens/ml_small/ratings.csv')
df.to_csv("../data/grouplens/ml_small/ratings_noh.csv", index=False, header=False) # 헤더 삭제
print(df.shape)
```

(100836, 4)

In [53]:

```
from surprise import Reader

reader = Reader(line_format='user item rating timestamp', sep=',', rating_scale=(0.5, 5))
data = Dataset.load_from_file('../data/grouplens/ml_small/ratings_noh.csv', reader=reader)
```

In [55]:

```
train, test = train_test_split(data,
                               test_size=.25, random_state=0)

print(type(train), type(test))
```

<class 'surprise.trainset.Trainset'> <class 'list'>

In [56]:



```
algo = SVD(n_factors=50, random_state=0)
algo.fit(train)
pred = algo.test(test)
accuracy.rmse(pred)
```

RMSE: 0.8682

Out [56]:

0.8681952927143516