**Programming Assignment #4 report**

한양대학교 컴퓨터전공

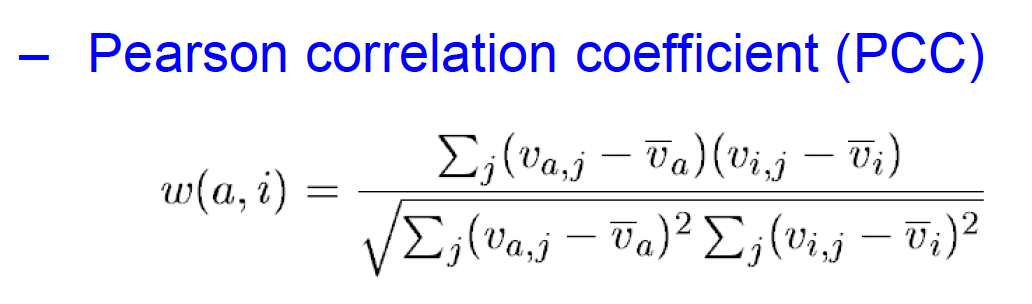
2012004087

이기준

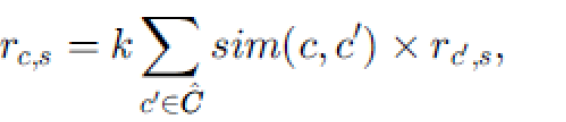
**Summary of your algorithm**

**Collaborative Filtering**

1. 유저 c의 선호하는 성향과 비슷한 이웃들의 그룹을 찾는다.



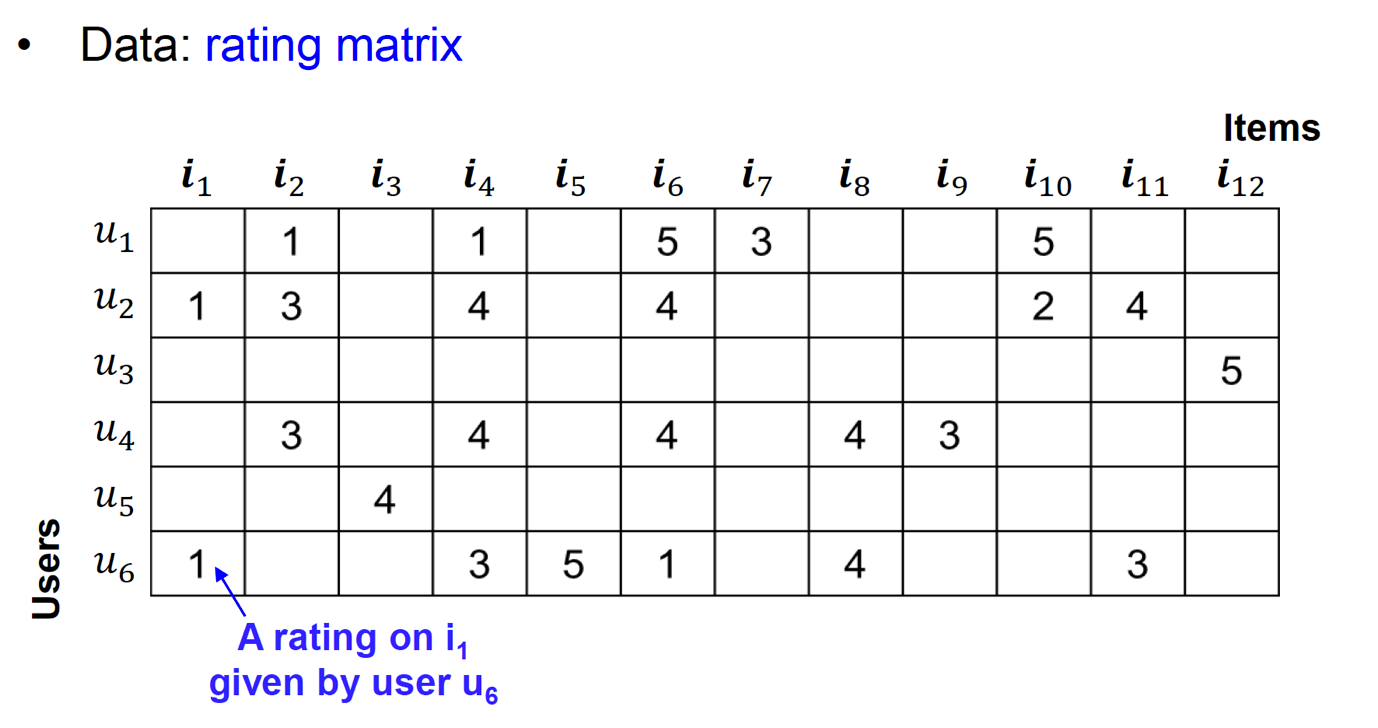
2. 유저 c의 이웃들의 평가를 기반으로 유저c를 위한 item s에 대한 평가를 한다.

****

**Detailed description of your codes (for each function)**

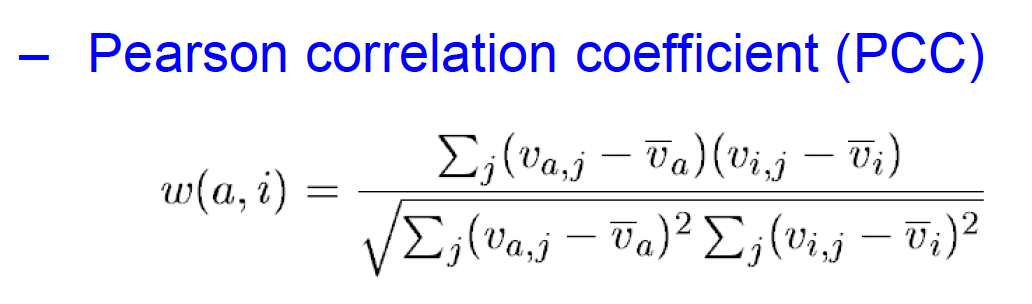
fileName = sys.argv[1]  
  
names = [**'user\_id'**, **'item\_id'**, **'rating'**, **'timestamp'**]  
df = pd.read\_csv(fileName, sep=**'\t'**, names=names)  
  
n\_users = df.user\_id.max()  
n\_items = df.item\_id.max()

ratings = np.zeros((n\_users, n\_items))  
test = np.zeros((n\_users, n\_items))  
**for** row **in** df.itertuples():  
 ratings[row[1]-1, row[2]-1] = row[3]

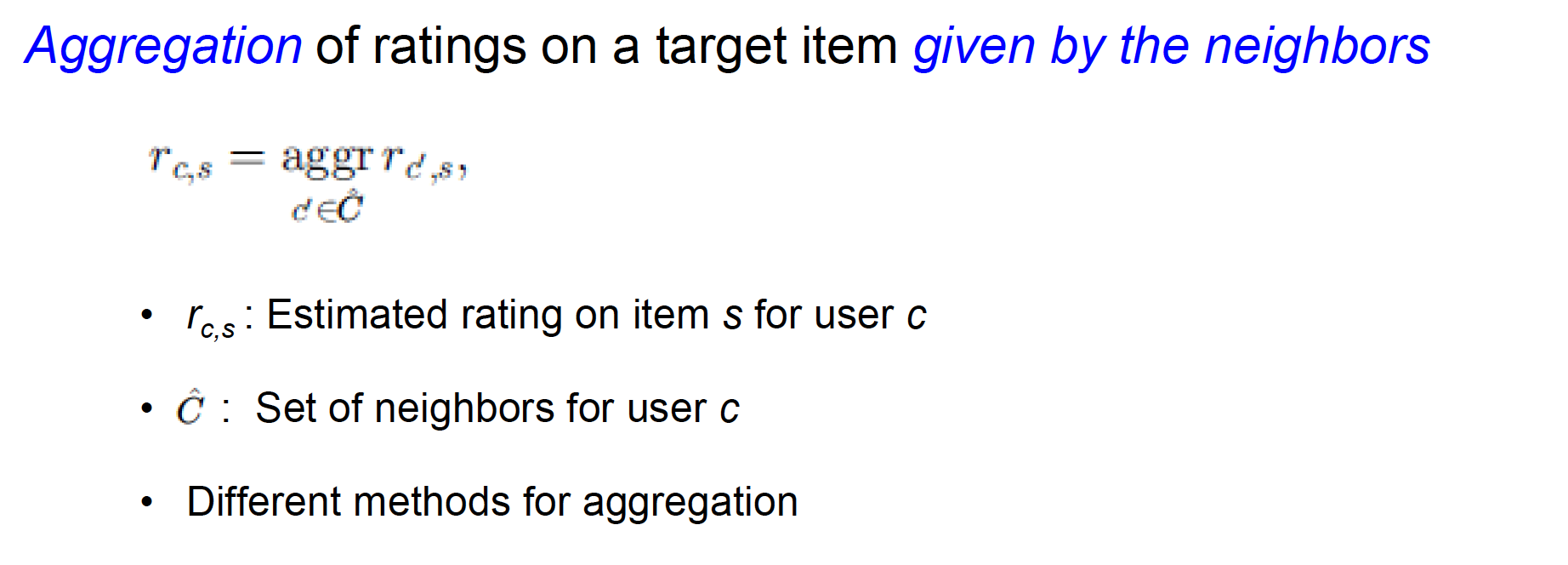
****

위의 그림처럼rating matrix를 만드는 코드이다.

**def** PCC(ratings, kind=**'user'**):  
 **if** kind == **'user'**:  
 axmax = 0  
 axmin = 1  
 sim = np.zeros((ratings.shape[axmax], ratings.shape[axmax]))  
 **for** u **in** range(ratings.shape[axmax]):  
 **for** uprime **in** range(u+1, ratings.shape[axmax]):  
 u\_mean = np.mean(ratings[u])  
 uprime\_mean = np.mean(ratings[uprime])  
  
 rui\_sqrd = 0.  
 ruprimei\_sqrd = 0.  
 **for** i **in** range(ratings.shape[axmin]):  
 sim[u, uprime] += (ratings[u, i] - u\_mean) \* (ratings[uprime, i] - uprime\_mean)  
 rui\_sqrd += (ratings[u, i] - u\_mean) \* (ratings[u, i] - u\_mean)  
 ruprimei\_sqrd += (ratings[uprime, i] - uprime\_mean) \* (ratings[uprime, i] - uprime\_mean)  
 sim[u, uprime] /= math.sqrt(rui\_sqrd \* ruprimei\_sqrd)  
  
 sim = sim +sim.T  
 **for** u **in** range(ratings.shape[axmax]):  
 sim[u,u] = 1  
  
 **return** sim

****

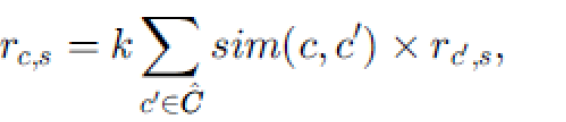
numberOfUser x numberOfUser similarity매트릭스가 생성된다.

****

**for** u **in** range(np.size(user\_similarity, 0)) :  
 **for** u\_prime **in** range(np.size(user\_similarity, 1)) :  
 **if** user\_similarity[u, u\_prime] < 0.2 :  
 user\_similarity[u, u\_prime] = .0

우선 similarity값이 0.2보다 작으면 그룹이 아니라고 간주, 0으로 설정한다**.**

**for** u **in** range(n\_users) :  
 print(u)  
 **for** C **in** range(np.size(user\_similarity, 0)):  
 **if**(user\_similarity[u, C] == 0) :  
 **continue  
 for** u\_prime **in** range(n\_items) :  
 test[u, u\_prime] += user\_similarity[u,C] \* ratings[C, u\_prime]

****

그 후 위의 식으로 평가하도록 하였다.

**Instructions for compiling your source codes at TA's computer (e.g. screenshot)**

OS : Windows10, 개발환경: pycharm 2017.2 사용언어:python (ver 3.6.4)

Numpy와 pandas를 사용하기 때문에 numpy와 pandas가 설치되어 있지않으면

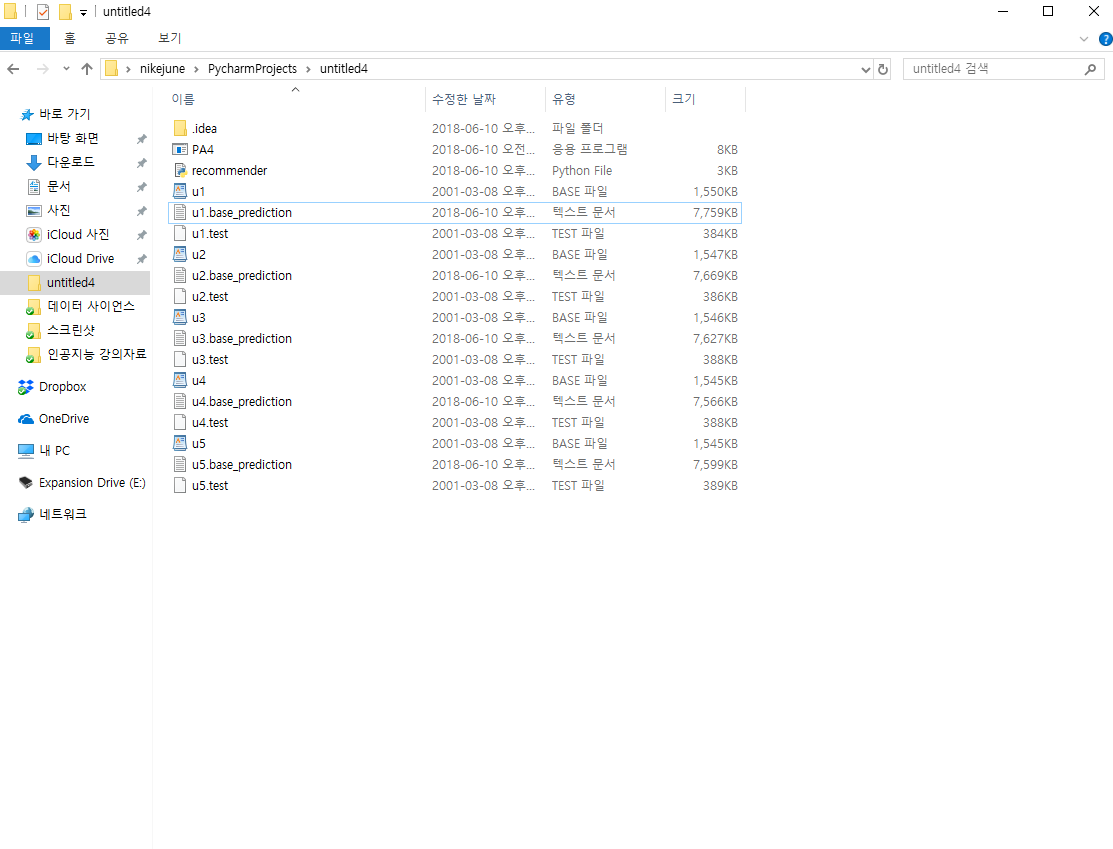
Pip3 install numpy

Pip3 install pandas

를 통해 설치를 해야한다.



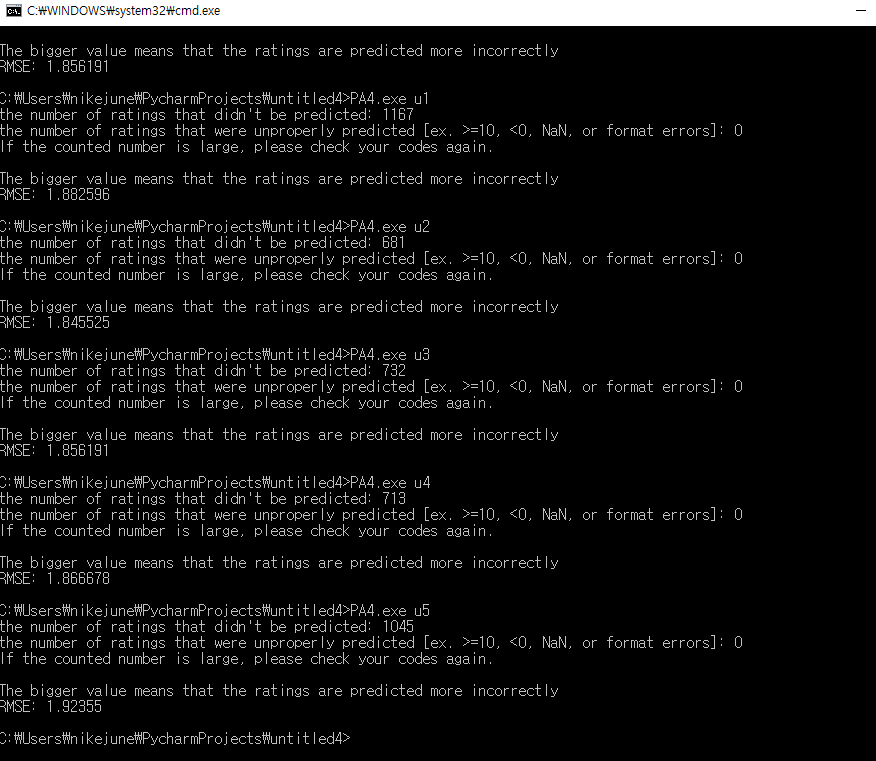
( PCC과정에서 거의 20-25분이 소요된다.)



**Any other specification of your implementation and testing**

**테스트결과**.

(윈도우10에서 실행)



예측하지 못한 rating은 700~1100

RMSE 값은 1.8~에서 1.9