In [257]:

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import mglearn
%matplotlib inline
import seaborn as sns
import platform
from matplotlib import font_manager , rc

if platform.system() == 'Darwin':
  rc('font' , family = 'AppleGothic')
elif platform.system() == 'Windows':
  path = 'C:/Windows/Fonts/malgun.ttf'
  font_name = font_manager.FontProperties(fname = path).get_name()
  rc('font' , family = font_name)
else:
  print('모름')
plt.rcParams['axes.unicode_minus'] = False
import warnings
warnings.filterwarnings('ignore')
```

executed in 24ms, finished 17:41:46 2023-10-30

1 데이터 셋 로딩과 주제 정의

In [258]:

```
data = pd.read_csv('speed_dating.csv')
```

executed in 44ms, finished 17:41:46 2023-10-30

In [259]:

```
data = data.iloc[:,2:]
```

executed in 15ms, finished 17:41:46 2023-10-30

In [260]:

```
data.columns
```

executed in 11ms, finished 17:41:46 2023-10-30

Out[260]:

```
Index(['gender', 'age', 'age_o', 'race', 'race_o', 'importance_same_race',
       'importance_same_religion', 'pref_o_attractive', 'pref_o_sincere',
       'pref_o_intelligence', 'pref_o_funny', 'pref_o_ambitious',
       'pref_o_shared_interests', 'attractive_o', 'sincere_o',
       'intelligence_o', 'funny_o', 'ambitous_o', 'shared_interests_o',
       'attractive_important', 'sincere_important', 'intellicence_important',
       'funny_important', 'ambtition_important', 'shared_interests_important',
       'attractive_partner', 'sincere_partner', 'intelligence_partner',
       'funny_partner', 'ambition_partner', 'shared_interests_partner',
       'interests_correlate', 'expected_happy_with_sd_people',
       'expected_num_interested_in_me', 'like', 'guess_prob_liked', 'met',
       'match'],
      dtype='object')
```

- perf_o_xxx : 상대방이 xxx 항목을 얼마나 중요시하는지에 대한 점수
- xxx_o : 상대방이 본인의 xxx 항목을 평가한 점수
- xxx_important : xxx 항목에 대해 본인이 얼마나 중요하게 생각하는지에 대한 점수
- xxx_partner : 본인이 상대방에 대한 xxx 항목 평가
- interests_correlate : 관심사 연관도
- expectd_happy_with_sd_people : 스피드 데이팅을 통해 만난 사람과 함께 할 때 , 얼마나 좋을지에 대한 기대치
- expected_num_interested_in_me : 얼마나 많은 사람이 나에게 관심을 보일지에 대한 기대치
- like : 파트너를 좋아하는지
- guss_prob_liked : 파트너가 나를 마음에 들어했을지에 대한 예상
- met : 이전에 만난 적이 있는지
- match : target

In [261]:

```
data.describe().T
```

executed in 79ms, finished 17:41:46 2023-10-30

Out[261]:

	count	mean	std	min	25%	50%	75%	max
age	8283.0	26.358928	3.566763	18.00	24.00	26.00	28.00	55.00
age_o	8274.0	26.364999	3.563648	18.00	24.00	26.00	28.00	55.00
importance_same_race	8299.0	3.784793	2.845708	0.00	1.00	3.00	6.00	10.00
importance_same_religion	8299.0	3.651645	2.805237	1.00	1.00	3.00	6.00	10.00
pref_o_attractive	8289.0	22.495347	12.569802	0.00	15.00	20.00	25.00	100.00
pref_o_sincere	8289.0	17.396867	7.044003	0.00	15.00	18.37	20.00	60.00
pref_o_intelligence	8289.0	20.270759	6.782895	0.00	17.39	20.00	23.81	50.00
pref_o_funny	8280.0	17.459714	6.085526	0.00	15.00	18.00	20.00	50.00
pref_o_ambitious	8271.0	10.685375	6.126544	0.00	5.00	10.00	15.00	53.00
pref_o_shared_interests	8249.0	11.845930	6.362746	0.00	9.52	10.64	16.00	30.00
attractive_o	8166.0	6.190411	1.950305	0.00	5.00	6.00	8.00	10.50
sincere_o	8091.0	7.175256	1.740575	0.00	6.00	7.00	8.00	10.00
intelligence_o	8072.0	7.369301	1.550501	0.00	6.00	7.00	8.00	10.00
funny_o	8018.0	6.400599	1.954078	0.00	5.00	7.00	8.00	11.00
ambitous_o	7656.0	6.778409	1.794080	0.00	6.00	7.00	8.00	10.00
shared_interests_o	7302.0	5.474870	2.156163	0.00	4.00	6.00	7.00	10.00
attractive_important	8299.0	22.514632	12.587674	0.00	15.00	20.00	25.00	100.00
sincere_important	8299.0	17.396389	7.046700	0.00	15.00	18.18	20.00	60.00
intellicence_important	8299.0	20.265613	6.783003	0.00	17.39	20.00	23.81	50.00
funny_important	8289.0	17.457043	6.085239	0.00	15.00	18.00	20.00	50.00
ambtition_important	8279.0	10.682539	6.124888	0.00	5.00	10.00	15.00	53.00
shared_interests_important	8257.0	11.845111	6.362154	0.00	9.52	10.64	16.00	30.00
attractive_partner	8176.0	6.189995	1.950169	0.00	5.00	6.00	8.00	10.00
sincere_partner	8101.0	7.175164	1.740315	0.00	6.00	7.00	8.00	10.00
intelligence_partner	8082.0	7.368597	1.550453	0.00	6.00	7.00	8.00	10.00
funny_partner	8028.0	6.400598	1.953702	0.00	5.00	7.00	8.00	10.00
ambition_partner	7666.0	6.777524	1.794055	0.00	6.00	7.00	8.00	10.00
shared_interests_partner	7311.0	5.474559	2.156363	0.00	4.00	6.00	7.00	10.00
interests_correlate	8220.0	0.196010	0.303539	-0.83	-0.02	0.21	0.43	0.91
expected_happy_with_sd_people	8277.0	5.534131	1.734059	1.00	5.00	6.00	7.00	10.00
expected_num_interested_in_me	1800.0	5.570556	4.762569	0.00	2.00	4.00	8.00	20.00
like	8138.0	6.134087	1.841285	0.00	5.00	6.00	7.00	10.00
guess_prob_liked	8069.0	5.207523	2.129565	0.00	4.00	5.00	7.00	10.00
met	8003.0	0.049856	0.282168	0.00	0.00	0.00	0.00	8.00
match	8378.0	0.164717	0.370947	0.00	0.00	0.00	0.00	1.00

In [262]:

```
#결측치 비율 확인
data.isna().mean()
```

executed in 14ms, finished 17:41:46 2023-10-30

Out[262]:

```
gender                         0.000000
age                            0.011339
age_o                          0.012413
race                           0.007520
race_o                         0.008713
importance_same_race           0.009429
importance_same_religion       0.009429
pref_o_attractive              0.010623
pref_o_sincere                 0.010623
pref_o_intelligence            0.010623
pref_o_funny                   0.011697
pref_o_ambitious               0.012772
pref_o_shared_interests        0.015397
attractive_o                   0.025304
sincere_o                      0.034256
intelligence_o                 0.036524
funny_o                        0.042970
ambitous_o                     0.086178
shared_interests_o             0.128432
attractive_important           0.009429
sincere_important              0.009429
intellicence_important         0.009429
funny_important                0.010623
ambtition_important            0.011817
shared_interests_important     0.014443
attractive_partner             0.024111
sincere_partner                0.033063
intelligence_partner           0.035331
funny_partner                  0.041776
ambition_partner               0.084984
shared_interests_partner       0.127357
interests_correlate            0.018859
expected_happy_with_sd_people  0.012055
expected_num_interested_in_me  0.785152
like                           0.028646
guess_prob_liked               0.036882
met                            0.044760
match                          0.000000
dtype: float64
```

- 종교와 인종 선호도 수치는 비어있을 경우 , 상관없음으로 간주 . 가중치를 곱할 때 1로 표기

In [263]:

```
data['importance_same_race'] = data['importance_same_race'].fillna(1)
```

executed in 13ms, finished 17:41:46 2023-10-30

In [264]:

```
data['importance_same_religion'] = data['importance_same_religion'].fillna(1)
```

executed in 14ms, finished 17:41:46 2023-10-30

In [265]:

```
data.info()
```

executed in 14ms, finished 17:41:46 2023-10-30

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8378 entries, 0 to 8377
Data columns (total 38 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   gender                         8378 non-null   object 
 1   age                            8283 non-null   float64
 2   age_o                          8274 non-null   float64
 3   race                           8315 non-null   object 
 4   race_o                         8305 non-null   object 
 5   importance_same_race           8378 non-null   float64
 6   importance_same_religion       8378 non-null   float64
 7   pref_o_attractive              8289 non-null   float64
 8   pref_o_sincere                 8289 non-null   float64
 9   pref_o_intelligence            8289 non-null   float64
 10  pref_o_funny                   8280 non-null   float64
 11  pref_o_ambitious               8271 non-null   float64
 12  pref_o_shared_interests        8249 non-null   float64
 13  attractive_o                   8166 non-null   float64
```

- 서로 평가를 해야하는데 , 평가에 기입 안한 것들 제거

In [266]:
```
data.dropna(subset = ['pref_o_attractive', 'pref_o_sincere', 'pref_o_intelligence',
       'pref_o_funny', 'pref_o_ambitious', 'pref_o_shared_interests',
       'attractive_o', 'sincere_o', 'intelligence_o', 'funny_o', 'ambitous_o',
       'shared_interests_o', 'attractive_important', 'sincere_important',
       'intellicence_important', 'funny_important', 'ambtition_important',
       'shared_interests_important', 'attractive_partner', 'sincere_partner',
       'intelligence_partner', 'funny_partner', 'ambition_partner',
       'shared_interests_partner'] , inplace = True)
```
executed in 12ms, finished 17:41:46 2023-10-30

In [267]:
```
data.info()
```
executed in 14ms, finished 17:41:46 2023-10-30

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5842 entries, 0 to 8377
Data columns (total 38 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   gender                         5842 non-null   object 
 1   age                            5826 non-null   float64
 2   age_o                          5826 non-null   float64
 3   race                           5842 non-null   object 
 4   race_o                         5842 non-null   object 
 5   importance_same_race           5842 non-null   float64
 6   importance_same_religion       5842 non-null   float64
 7   pref_o_attractive              5842 non-null   float64
 8   pref_o_sincere                 5842 non-null   float64
 9   pref_o_intelligence            5842 non-null   float64
 10  pref_o_funny                   5842 non-null   float64
 11  pref_o_ambitious               5842 non-null   float64
 12  pref_o_shared_interests        5842 non-null   float64
 13  attractive_o                   5842 non-null   float64
 14  sincere_o                      5842 non-null   float64
 15  intelligence_o                 5842 non-null   float64
 16  funny_o                        5842 non-null   float64
 17  ambitous_o                     5842 non-null   float64
 18  shared_interests_o             5842 non-null   float64
 19  attractive_important           5842 non-null   float64
 20  sincere_important              5842 non-null   float64
 21  intellicence_important         5842 non-null   float64
 22  funny_important                5842 non-null   float64
 23  ambtition_important            5842 non-null   float64
 24  shared_interests_important     5842 non-null   float64
 25  attractive_partner             5842 non-null   float64
 26  sincere_partner                5842 non-null   float64
 27  intelligence_partner           5842 non-null   float64
 28  funny_partner                  5842 non-null   float64
 29  ambition_partner               5842 non-null   float64
 30  shared_interests_partner       5842 non-null   float64
 31  interests_correlate            5842 non-null   float64
 32  expected_happy_with_sd_people  5826 non-null   float64
 33  expected_num_interested_in_me  1260 non-null   float64
 34  like                           5823 non-null   float64
 35  guess_prob_liked               5786 non-null   float64
 36  met                            5716 non-null   float64
 37  match                          5842 non-null   int64  
dtypes: float64(34), int64(1), object(3)
memory usage: 1.7+ MB
```

- expected_num_interested_in_me와 guess_prob_liked는 상대방이 본인을 어떻게 생각하느냐에 대한 '예상' 이므로 , 제거

In [268]:
```
data.drop(['guess_prob_liked' , 'expected_num_interested_in_me'] , axis = 1 , inplace = True)
```
executed in 12ms, finished 17:41:46 2023-10-30

- expected_happy_with_sd_people는 이사람이랑 만나면 행복할 수 있을까? 라는 기대치이다. 이것은 서로의 점수를 통해서 예측할 수 있으므로 제거

In [269]:
```
data.drop('expected_happy_with_sd_people' , axis = 1 , inplace = True)
```
executed in 14ms, finished 17:41:46 2023-10-30

- like도 서로의 점수를 통해서 예측할 수 있으므로 제거

In [270]:
```
data.drop('like' , axis = 1 , inplace = True)
```
executed in 14ms, finished 17:41:46 2023-10-30

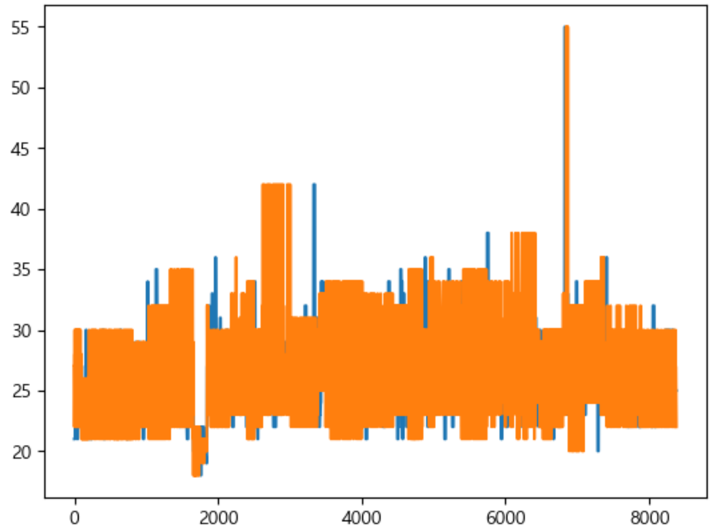
- met 은 만난 적이 있는지에 대한 0 or 1이다. 기입을 안한 건 만나지 않았다는 것으로 간주. 0으로 채우기

In [271]:
```
data.met = data.met.fillna(0)
```
executed in 15ms, finished 17:41:46 2023-10-30

In [272]:
```
plt.plot(data.age)
plt.plot(data.age_o)
```

executed in 169ms, finished 17:41:47 2023-10-30

Out[272]:
```
[<matplotlib.lines.Line2D at 0x15b0d2b3760>]
```



In [273]:
```
data[data.age.isna()]
```

executed in 46ms, finished 17:41:47 2023-10-30

Out[273]:

	gender	age	age_o	race	race_o	importance_same_race	importance_same_religion	pref_o_attractive	pref_o_si
7476	female	NaN	25.0	European/Caucasian-American	Asian/PacificIslander/Asian-American	1.0	1.0	15.0	
7477	female	NaN	26.0	European/Caucasian-American	Asian/PacificIslander/Asian-American	1.0	1.0	25.0	
7479	female	NaN	24.0	European/Caucasian-American	Other	1.0	1.0	30.0	
7480	female	NaN	23.0	European/Caucasian-American	European/Caucasian-American	1.0	1.0	23.0	
7481	female	NaN	29.0	European/Caucasian-American	European/Caucasian-American	1.0	1.0	30.0	
7482	female	NaN	22.0	European/Caucasian-American	Asian/PacificIslander/Asian-American	1.0	1.0	20.0	
7484	female	NaN	22.0	European/Caucasian-American	European/Caucasian-American	1.0	1.0	30.0	
7486	female	NaN	23.0	European/Caucasian-American	Asian/PacificIslander/Asian-American	1.0	1.0	40.0	
7487	female	NaN	23.0	European/Caucasian-American	Asian/PacificIslander/Asian-American	1.0	1.0	25.0	
7488	female	NaN	24.0	European/Caucasian-American	Asian/PacificIslander/Asian-American	1.0	1.0	20.0	
7489	female	NaN	23.0	European/Caucasian-American	Asian/PacificIslander/Asian-American	1.0	1.0	15.0	
7490	female	NaN	24.0	European/Caucasian-American	European/Caucasian-American	1.0	1.0	20.0	
7491	female	NaN	30.0	European/Caucasian-American	European/Caucasian-American	1.0	1.0	35.0	
7492	female	NaN	30.0	European/Caucasian-American	European/Caucasian-American	1.0	1.0	25.0	
7494	female	NaN	28.0	European/Caucasian-American	European/Caucasian-American	1.0	1.0	20.0	
7495	female	NaN	30.0	European/Caucasian-American	European/Caucasian-American	1.0	1.0	30.0	

16 rows × 34 columns

In [274]:
```
data[data.age_o.isna()]
```
executed in 45ms, finished 17:41:47 2023-10-30

Out[274]:

	gender	age	age_o	race	race_o	importance_same_race	importance_same_religion	pref_o_attractive	pref_o_si
7897	male	25.0	NaN	Asian/PacificIslander/Asian-American	European/Caucasian-American	5.0	1.0	20.0	
7919	male	26.0	NaN	Asian/PacificIslander/Asian-American	European/Caucasian-American	8.0	3.0	20.0	
7963	male	24.0	NaN	Other	European/Caucasian-American	1.0	1.0	20.0	
7985	male	23.0	NaN	European/Caucasian-American	European/Caucasian-American	5.0	6.0	20.0	
8007	male	29.0	NaN	European/Caucasian-American	European/Caucasian-American	1.0	1.0	20.0	
8029	male	22.0	NaN	Asian/PacificIslander/Asian-American	European/Caucasian-American	7.0	1.0	20.0	
8073	male	22.0	NaN	European/Caucasian-American	European/Caucasian-American	6.0	6.0	20.0	
8117	male	23.0	NaN	Asian/PacificIslander/Asian-American	European/Caucasian-American	2.0	2.0	20.0	
8139	male	23.0	NaN	Asian/PacificIslander/Asian-American	European/Caucasian-American	7.0	1.0	20.0	
8161	male	24.0	NaN	Asian/PacificIslander/Asian-American	European/Caucasian-American	9.0	6.0	20.0	
8183	male	23.0	NaN	Asian/PacificIslander/Asian-American	European/Caucasian-American	3.0	8.0	20.0	
8205	male	24.0	NaN	European/Caucasian-American	European/Caucasian-American	7.0	1.0	20.0	
8227	male	30.0	NaN	European/Caucasian-American	European/Caucasian-American	3.0	4.0	20.0	
8249	male	30.0	NaN	European/Caucasian-American	European/Caucasian-American	1.0	1.0	20.0	
8293	male	28.0	NaN	European/Caucasian-American	European/Caucasian-American	2.0	3.0	20.0	
8315	male	30.0	NaN	European/Caucasian-American	European/Caucasian-American	5.0	6.0	20.0	

16 rows × 34 columns

- 나이를 기재하지 않은 32개의 NaN값을 보면 , 16명의 여자가 기재하지 않았다는 것을 알 수 있다. 그런데 인종까지 같다. 한명인가??

- 16개의 NaN값은 여자의 나이의 평균으로 대체하자. 나이의 분포를 보면 55세만 아니면 중요하지 않을 것 같다.

In [275]:
```
female = data.loc[data['gender'] == 'female' , 'age'].mean()
```
executed in 14ms, finished 17:41:47 2023-10-30

In [276]:
```
data.fillna(female , inplace = True)
```
executed in 12ms, finished 17:41:47 2023-10-30

- pref와 19열부터 나오는 important들은 각각 6개의 컬럼이며 , 그 합은 100이다. 가중치를 둘 때 , 퍼센트로 바꾸고 , 실질적인 점수에 가중치를 곱해보자

In [277]:
```
data.columns
```
executed in 15ms, finished 17:41:47 2023-10-30

Out[277]:
```
Index(['gender', 'age', 'age_o', 'race', 'race_o', 'importance_same_race',
       'importance_same_religion', 'pref_o_attractive', 'pref_o_sincere',
       'pref_o_intelligence', 'pref_o_funny', 'pref_o_ambitious',
       'pref_o_shared_interests', 'attractive_o', 'sincere_o',
       'intelligence_o', 'funny_o', 'ambitous_o', 'shared_interests_o',
       'attractive_important', 'sincere_important', 'intellicence_important',
       'funny_important', 'ambtition_important', 'shared_interests_important',
       'attractive_partner', 'sincere_partner', 'intelligence_partner',
       'funny_partner', 'ambition_partner', 'shared_interests_partner',
       'interests_correlate', 'met', 'match'],
      dtype='object')
```

In [278]:
```
data[['pref_o_attractive', 'pref_o_sincere',
      'pref_o_intelligence', 'pref_o_funny', 'pref_o_ambitious',
      'pref_o_shared_interests']] = data[['pref_o_attractive', 'pref_o_sincere',
      'pref_o_intelligence', 'pref_o_funny', 'pref_o_ambitious',
      'pref_o_shared_interests']]/100
```
executed in 15ms, finished 17:41:47 2023-10-30

In [279]:
```
data[['attractive_important', 'sincere_important', 'intellicence_important',
'funny_important', 'ambtition_important', 'shared_interests_important']] = data[['attractive_important', 'sincere_important', 'intelli
       'funny_important', 'ambtition_important', 'shared_interests_important']]/100
```

executed in 14ms, finished 17:41:47 2023-10-30

In [280]:
```
for i , j in enumerate(['attractive_o', 'sincere_o', 'intellicence_o','funny_o', 'ambtition_o', 'shared_interests_o']):
    data[j] = data[['pref_o_attractive', 'pref_o_sincere',
       'pref_o_intelligence', 'pref_o_funny', 'pref_o_ambitious',
       'pref_o_shared_interests']].iloc[:,i] * data[['attractive_o', 'sincere_o',
       'intelligence_o', 'funny_o', 'ambitous_o', 'shared_interests_o']].iloc[:,i]
```

executed in 29ms, finished 17:41:47 2023-10-30

In [281]:
```
for i,j in enumerate(['attractive', 'sincere', 'intellicence','funny', 'ambtition', 'shared_interests']):
    data[j] = data[['attractive_important', 'sincere_important', 'intellicence_important',
'funny_important', 'ambtition_important', 'shared_interests_important']].iloc[:,i] * data[['attractive_partner', 'sincere_partner', 'i
       'funny_partner', 'ambition_partner', 'shared_interests_partner']].iloc[:,i]
```

executed in 30ms, finished 17:41:47 2023-10-30

In [282]:
```
data.drop(['attractive_important', 'sincere_important', 'intellicence_important',
'funny_important', 'ambtition_important', 'shared_interests_important','pref_o_attractive', 'pref_o_sincere',
       'pref_o_intelligence', 'pref_o_funny', 'pref_o_ambitious',
       'pref_o_shared_interests'] , axis = 1 , inplace = True)
```

executed in 13ms, finished 17:41:47 2023-10-30

In [283]:
```
data.head()
```

executed in 30ms, finished 17:41:47 2023-10-30

Out[283]:

	gender	age	age_o	race	race_o	importance_same_race	importance_same_religion	attractive_o	sincere_o	i
0	female	21.0	27.0	Asian/PacificIslander/Asian-American	European/Caucasian-American	2.0	4.0	2.1	1.6	
1	female	21.0	22.0	Asian/PacificIslander/Asian-American	European/Caucasian-American	2.0	4.0	4.2	0.0	
2	female	21.0	22.0	Asian/PacificIslander/Asian-American	Asian/PacificIslander/Asian-American	2.0	4.0	1.9	1.8	
3	female	21.0	23.0	Asian/PacificIslander/Asian-American	European/Caucasian-American	2.0	4.0	2.1	0.4	
4	female	21.0	24.0	Asian/PacificIslander/Asian-American	Latino/HispanicAmerican	2.0	4.0	2.4	0.7	

5 rows × 30 columns

In [284]:
```
data.columns
```

executed in 11ms, finished 17:41:47 2023-10-30

Out[284]:
```
Index(['gender', 'age', 'age_o', 'race', 'race_o', 'importance_same_race',
       'importance_same_religion', 'attractive_o', 'sincere_o',
       'intelligence_o', 'funny_o', 'ambitous_o', 'shared_interests_o',
       'attractive_partner', 'sincere_partner', 'intelligence_partner',
       'funny_partner', 'ambition_partner', 'shared_interests_partner',
       'interests_correlate', 'met', 'match', 'intellicence_o', 'ambtition_o',
       'attractive', 'sincere', 'intellicence', 'funny', 'ambtition',
       'shared_interests'],
      dtype='object')
```

In [285]:
```
data.drop(['attractive_o', 'sincere_o',
       'intelligence_o', 'funny_o', 'ambitous_o', 'shared_interests_o',
       'attractive_partner', 'sincere_partner', 'intelligence_partner',
       'funny_partner', 'ambition_partner', 'shared_interests_partner'] , axis = 1 , inplace = True)
```

executed in 29ms, finished 17:41:47 2023-10-30

In [286]:
```
data.columns
```

executed in 14ms, finished 17:41:47 2023-10-30

Out[286]:
```
Index(['gender', 'age', 'age_o', 'race', 'race_o', 'importance_same_race',
       'importance_same_religion', 'interests_correlate', 'met', 'match',
       'intellicence_o', 'ambtition_o', 'attractive', 'sincere',
       'intellicence', 'funny', 'ambtition', 'shared_interests'],
      dtype='object')
```

In [287]:
```
data = data[['gender', 'age', 'age_o', 'race', 'race_o', 'importance_same_race',
       'importance_same_religion', 'interests_correlate', 'met',
       'intellicence_o', 'ambtition_o', 'attractive', 'sincere',
       'intellicence', 'funny', 'ambtition', 'shared_interests','match']]
```

executed in 14ms, finished 17:41:47 2023-10-30

- age의 결측치 처리 과정 중 , 남녀 두 경우에 대한 매치가 중복되므로 , gender는 분석에 필요가 없다고 판단. 제거

In [288]:
```
data.drop('gender' , axis = 1 , inplace = True)
```
executed in 13ms, finished 17:41:47 2023-10-30

- 인종에 대한 수치는 importance_same_race에서 다루므로 , 인종도 제거

In [289]:
```
data.drop(['race','race_o'] , axis = 1 , inplace = True)
```
executed in 14ms, finished 17:41:47 2023-10-30

In [290]:
```
data.interests_correlate
```
executed in 17ms, finished 17:41:47 2023-10-30

Out[290]:
```
0       0.14
1       0.54
2       0.16
3       0.61
4       0.21
        ...
8367    0.37
8368    0.27
8369    0.45
8370    0.35
8377    0.01
Name: interests_correlate, Length: 5842, dtype: float64
```

In [291]:
```
data.head()
```
executed in 24ms, finished 17:41:47 2023-10-30

Out[291]:

	age	age_o	importance_same_race	importance_same_religion	interests_correlate	met	intellicence_o	ambtition_o	attractive	sincere	intellicence	f
0	21.0	27.0	2.0	4.0	0.14	0.0	1.60	0.00	0.90	1.8	1.4	
1	21.0	22.0	2.0	4.0	0.54	1.0	0.00	0.00	1.05	1.6	1.4	
2	21.0	22.0	2.0	4.0	0.16	1.0	1.90	1.40	0.75	1.6	1.8	
3	21.0	23.0	2.0	4.0	0.61	0.0	1.35	0.45	1.05	1.2	1.6	
4	21.0	24.0	2.0	4.0	0.21	0.0	1.80	0.90	0.75	1.2	1.4	

In [292]:
```
data.columns
```
executed in 13ms, finished 17:41:47 2023-10-30

Out[292]:
```
Index(['age', 'age_o', 'importance_same_race', 'importance_same_religion',
       'interests_correlate', 'met', 'intellicence_o', 'ambtition_o',
       'attractive', 'sincere', 'intellicence', 'funny', 'ambtition',
       'shared_interests', 'match'],
      dtype='object')
```

In [293]:
```
hap = data['importance_same_race'] + data['importance_same_religion']
```
executed in 14ms, finished 17:41:47 2023-10-30

In [294]:
```
data.drop('met' , axis = 1 , inplace = True)
```
executed in 12ms, finished 17:41:47 2023-10-30

In [295]:
```
data1 = data.iloc[:,:-1].to_numpy()
target = data.iloc[:,-1].to_numpy()
```
executed in 13ms, finished 17:41:47 2023-10-30

In [296]:
```
data.match.value_counts()
```
executed in 14ms, finished 17:41:47 2023-10-30

Out[296]:
```
0    4802
1    1040
Name: match, dtype: int64
```

In [297]:
```
from sklearn.model_selection import train_test_split

train_input , test_input , train_target , test_target = train_test_split(data1 , target , test_size = 0.2 , stratify = target)
```
executed in 15ms, finished 17:41:47 2023-10-30

In [298]:
```
mport accuracy_score , precision_score , recall_score , roc_auc_score , f1_score , confusion_matrix , roc_curve , precision_recall_cur
```
executed in 13ms, finished 17:41:47 2023-10-30

In [299]:
```
from xgboost import XGBClassifier
xgb = XGBClassifier(n_estimators = 400)
xgb.fit(train_input , train_target)

xgb_pred = xgb.predict(test_input)

accuracy_score(test_target , xgb_pred)
```
executed in 512ms, finished 17:41:48 2023-10-30

Out[299]:
```
0.8092386655260907
```

In [300]:
```
from sklearn.model_selection import GridSearchCV

params = {'n_estimators' : [200,300,400,500,600,700,800],
         'learning_rate' : [0.2,0.3,0.4,0.5,0.6,0.7]}
```
executed in 14ms, finished 17:41:48 2023-10-30

In [301]:
```
gs = GridSearchCV(XGBClassifier(random_state = 42) , params , n_jobs = -1)
gs.fit(train_input , train_target)
```
executed in 1m 10.6s, finished 17:42:58 2023-10-30

Out[301]:
```
▸       GridSearchCV
▸ estimator: XGBClassifier
        ▸ XGBClassifier
```

In [302]:
```
gs.best_params_
```
executed in 14ms, finished 17:42:58 2023-10-30

Out[302]:
```
{'learning_rate': 0.2, 'n_estimators': 200}
```

In [303]:
```
xgb2 = XGBClassifier(n_estimators = 200 , learning_rate = 0.2 , random_state = 42)
xgb2.fit(train_input , train_target)
```
executed in 493ms, finished 17:42:59 2023-10-30

Out[303]:
```
▾                          XGBClassifier
          colsample_bylevel=None, colsample_bynode=None,
          colsample_bytree=None, early_stopping_rounds=None,
          enable_categorical=False, eval_metric=None, feature_types=None,
          gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
          interaction_constraints=None, learning_rate=0.2, max_bin=None,
          max_cat_threshold=None, max_cat_to_onehot=None,
          max_delta_step=None, max_depth=None, max_leaves=None,
          min_child_weight=None, missing=nan, monotone_constraints=None,
          n_estimators=200, n_jobs=None, num_parallel_tree=None,
          predictor=None, random_state=42, ...)
```

In [304]:
```
accuracy_score(test_target , xgb2.predict(test_input)), precision_score(test_target , xgb2.predict(test_input))
```
executed in 28ms, finished 17:42:59 2023-10-30

Out[304]:
```
(0.8212147134302823, 0.49473684210526314)
```

In [305]:
```
confusion_matrix(test_target , xgb2.predict(test_input))
```
executed in 13ms, finished 17:42:59 2023-10-30

Out[305]:
```
array([[913,  48],
       [161,  47]], dtype=int64)
```

- 실제값은 만났는데 , 만나지 않았다고 예측한 수가 너무 많다.

In [306]:
```
xgb2.feature_importances_
```
executed in 13ms, finished 17:42:59 2023-10-30

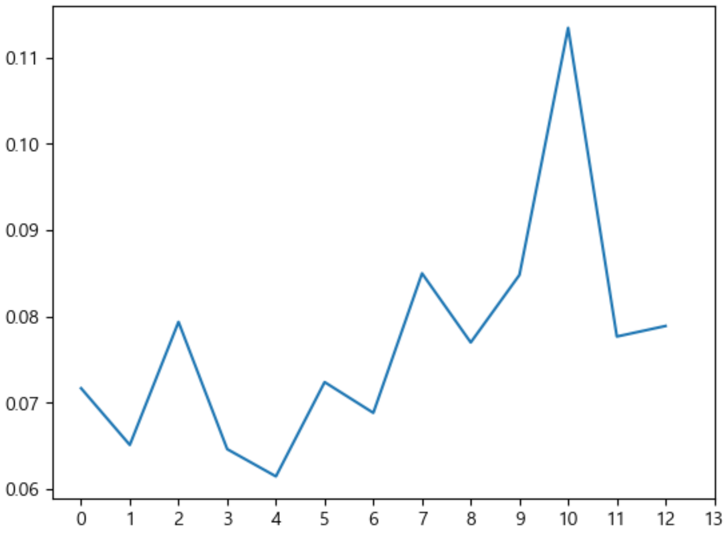
Out[306]:
```
array([0.07165853, 0.06509665, 0.07934473, 0.06460647, 0.06145023,
       0.07237166, 0.06881104, 0.08496993, 0.07696783, 0.08478513,
       0.1134124 , 0.07765094, 0.0788744 ], dtype=float32)
```

In [307]:
```
len(data.columns)
```
executed in 13ms, finished 17:42:59 2023-10-30

Out[307]:
```
14
```

In [308]:
```
plt.plot(xgb2.feature_importances_)
plt.xticks(np.arange(14) , np.arange(0,14,1))
plt.show()
```

executed in 169ms, finished 17:42:59 2023-10-30



In [309]:
```
data.columns.tolist()[10]
```

executed in 14ms, finished 17:42:59 2023-10-30

Out[309]:
```
'funny'
```

- 이 모델에서 가장 중요한 것은 상대방이 얼마나 재밌는지에 대한 점수가 가장 중요했다고 볼 수 있다.
- 그런데 점수가 너무 낮다..