

```
In [3]: cd Desktop

[Errno 2] No such file or directory: 'Desktop'
/Users/jeongmingi/Desktop
```

```
In [8]: from sklearn.externals import joblib
filename = "age_my_model_70.pkl"
clf = joblib.load(filename)
```

```
In [32]: cd data

/Users/jeongmingi/Desktop/data
```

```
In [33]: import pickle
uapp = pickle.load(file('user_app.df'))
```

```
In [34]: import numpy as np
```

```
In [38]: cd DataMining/pickles

/Users/jeongmingi/Desktop/DataMining/pickles
```

```
In [41]: EF_table = pickle.load(file('entityid_featureid.pkl'))
```

```
In [42]: def what_ages(i):
    user_ = uapp_.ix[(uapp_['user_id'] == i)]
    #user_ = uapp_[uapp_.apply(lambda x: x['user_id'] == i, axis=1)]
    ##### pickle 'entityid_featureid' used for translating...####
    #####
    X_test = (0 , list(user_.entity_id))
    temp = []
    for ii in X_test[1]:
        c_i = str(ii)
        if(EF_table.has_key(c_i) == True):
            temp.append(EF_table[c_i])
        else:
            pass
            #print(i , False)
    #####
    ##### vectorization... It will have 52,600 features and not ha
    ##### weight... ㅎㅎㅎ 그러나 0~1의 가중치를 줄 수 있지 않을까jQuery171
    n_features = 49683 ##!!!!!!
    temp2 = np.zeros(n_features, dtype = np.float64)
    for ii in range(n_features):
        if((ii+1) in temp):
            temp2[ii] = temp2[ii] + 1
        else:
            temp2[ii] = 0
    X_test = np.array([temp2])
    ages = clf.predict(X_test)
    #print int(gender)
    #X_test = np.array([0, temp])
```

```
return i, int(ages)
```

```
In [43]: ##### but, uapp으로부터 uapp_에 user_id를 추가하는 것 외에....
##### from Profiled_, gender를 먼저 수행한다고 가정하면 uapp이 아닌 'profiled_'
##### gender가 결정된 정보에 연령대(ages)를 추가할 것이다.

uapp_ = uapp[['user_id', 'entity_id']]
user_id_ = uapp_['user_id'].unique()
```

```
In [44]: user = []
ages = []
user_id__ = user_id[:1000]
for i in user_id__:
    temp1, temp2 = what_ages(i)
    user.append(temp1)
    ages.append(temp2)
```

```
In [47]: uapp["ages"] = 0
```

```
In [48]: for i, uid in enumerate(user_id__):
# print(i, uid)
uapp.ix[ (uapp['user_id'] == uid), 'ages' ] = ages[i]
```

```
In [49]: t = uapp[25:35].copy()
t.pop('usage')
t
```

Out[49]:

	create_date	update_date	user_id	entity_id	is_deleted	ages
25	20130317015228	20130730090416	3023116	1599923	True	4
26	20130222103311	20130728184307	3246192	1693135	True	1
27	20130310141222	20130729230224	1758743	1793010	True	2
28	20130317015228	20130730090416	3023116	1541687	True	4
29	20130412063020	20130730090416	3023116	1777213	True	4
30	20130607151638	20130728193812	2115990	1854907	True	1
31	20130331040556	20130731075255	4305863	1612457	True	1
32	20130625120908	20130729230224	1758743	1864689	True	2
33	20130604154438	20130730054858	3611801	1841672	True	1
34	20130317015228	20130730090416	3023116	1779808	True	4

```
In [ ]: #####
```

```
#####
##### easy shearch user id, qender, aqes... #####
```

```
In [41]: ls
558691_589646044379538_942129678_n-1.jpg
DataMining/
DataMining2/
NPKI/
Untitled0.ipynb
Untitled1.ipynb
Untitled2.ipynb
Untitled3.ipynb
adme_analysis_temp.ipynb
age_my_model_70.pkl
document_classification.ipynb
entityid_featureid.pkl
my_model.pkl
my_model10000.pkl
new_clustering.ipynb
predicted_user_apps.ipynb
profiled_user_gender.pkl
user_app_predicted.df
user_gender 2.txt
user_gender 3.txt
user_gender 4.txt
user_gender 5.txt
user_gender.txt
user_gender7_pred
```

```
In [51]: import pickle
users= pickle.load(file('profiled_user_gender.pkl'))
```

```
In [52]: users["ages"] = 0
```

```
In [50]:
```

```
In [ ]:
```

```
In [ ]: ##### SUMMARY OF DATAFRAME #####
```

```
In [67]: for i, uid in enumerate(users['user']):
          #print(i, uid)
          temp, age = what_ages(uid)
          users.ix[ (users['user'] == uid), 'ages' ] = age
```

```
Out[67]: [2115990, 2485217, 3023116, 4305863, 3611801]
```

```
In [ ]:
```

```
In [73]: users[10:20]
```

```
Out[73]:
```

	gender	user	ages
10	1	2115990	1
11	2	2485217	1
12	1	3023116	4
13	2	4305863	1
14	1	3611801	1
15	1	4828587	1
16	2	2086263	3
17	2	4851916	1
18	2	5100778	1
19	2	2686098	1

```
In [74]: import pickle
filename="profiled_user_gender_ages.pkl"
fout = file(filename,"w")
pickle.dump(users,fout)
```

```
In [ ]:
```