In [35]:
```
cd
```
```
/Users/jeongmingi
```

In [22]:
```
cd Desktop/DataMining/pickles
```
```
/Users/jeongmingi/Desktop/DataMining/pickles
```

In [23]:
```
import pickle
EF_table = pickle.load(file('entityid_featureid.pkl'))
```

In [32]:
```
cd Desktop/DataMining/studied
```
```
/Users/jeongmingi/Desktop/DataMining/studied
```

In [33]:
```
from sklearn.externals import joblib
filename = "my_model10000.pkl"
clf = joblib.load(filename)
```

In [39]:
```
cd desktop
```
```
/Users/jeongmingi/Desktop
```

In [40]:
```
uapp = pickle.load(file('data/user_app.df'))
```

In [7]:
```
type(uapp)
```

Out[7]:
```
pandas.core.frame.DataFrame
```

In [41]:
```
import numpy as np
```

In [9]:
```
##too take long times...'''
##
##
#user_id = uapp['user_id'].unique()
#user_id_ = user_id[:1]
#user_id_
#for i in user_id_:
#    user = uapp[uapp.apply(lambda x: x['user_id'] == i, axis=1)]
```

In [42]:
```
uapp_ = uapp[['user_id', 'entity_id']]
user_id_ = uapp_['user_id'].unique()
```

In [46]:
```
cd DataMining/pickles
```
```
/Users/jeongmingi/Desktop/DataMining/pickles
```

In [9]:
```
ls
```

```
entityid_featureid.pkl          profiled_user_gender_ages.pkl
profiled_user_gender.pkl
```

In [47]:
```
EF_table = pickle.load(file('entityid_featureid.pkl'))
```

In [48]:
```
def what_gender(i):
    user_ = uapp_.ix[(uapp_['user_id'] == i)]
    #user_ = uapp_[uapp_.apply(lambda x: x['user_id'] == i, axis=1)]
    ########## pickle 'entityid_featureid' uesed 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)
    ###########################################################################
    n_features = 52600
    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])
    gender = clf.predict(X_test)
    #print int(gender)
    #X_test = np.array([0, temp])
    return i, int(gender)
```

In [14]:
```
```

In [49]:
```
user = []
gender = []
user_id__ = user_id_[:1000]
for i in user_id__:
    temp1, temp2 = what_gender(i)
    user.append(temp1)
    gender.append(temp2)
```

In [53]:
```
```

In [365]:
```
#entityL = [ 1612462, 1651419 ]
#for e in entityL:
#    uapp.ix[ (uapp['user_id'] == 2007318) & (uapp['entity_id'] == e),
```

In [50]:
```
print (user[90:100], gender[90:100])
```

```
([1502177, 1280465, 1533210, 3879291, 11388, 3342187, 356897, 5104828,
5473218, 2831106], [2, 1, 2, 2, 2, 2, 2, 2, 2, 1])
```

In [51]:
```
count_male = 0;
count_female = 0;
for i in gender:
    if i==1:
        count_male = count_male+1
    elif i==2:
        count_female = count_female+1
print count_male, count_female
```

```
219 781
```

In [49]:
```
#temp = {'gender' : 0}
```

In [65]:
```
#uapp.ix[ (uapp['user_id'] == 2007318), 'gender' ] = 0
```

In [52]:
```
uapp["gender"] = 0
```

In [53]:
```
gender[:10]
```

Out[53]:
```
[2, 2, 2, 2, 2, 2, 1, 2, 1, 2]
```

In [54]:
```
for i, uid in enumerate(user_id__):
    print(i, uid)
    uapp.ix[ (uapp['user_id'] == uid), 'gender' ] = gender[i]
#uapp[:10]
```

```
(0, 2007318)
(1, 5188098)
(2, 5207032)
(3, 5183947)
(4, 1758743)
(5, 5062327)
(6, 3046239)
(7, 2382088)
(8, 1138375)
(9, 3246192)
(10, 2115990)
(11, 2485217)
(12, 3023116)
(13, 4305863)
(14, 3611801)
(15, 4828587)
(16, 2086263)
(17, 4851916)
(18, 5100778)
(19, 2686098)
```

In [60]:
```
t = uapp[20:30].copy()
```

```
t.pop('usage')
```

Out[60]:

	create_date	update_date	user_id	entity_id	is_deleted	gender
20	20130419110256	20130730204557	1138375	1522477	True	1
21	20130222103311	20130728184307	3246192	1524458	True	2
22	20130222131415	20130728193812	2115990	1605156	True	1
23	20130225002158	20130729035609	2485217	1675125	True	2
24	20130512120918	20130730090416	3023116	1506864	True	1
25	20130317015228	20130730090416	3023116	1599923	True	1
26	20130222103311	20130728184307	3246192	1693135	True	2
27	20130310141222	20130729230224	1758743	1793010	True	2
28	20130317015228	20130730090416	3023116	1541687	True	1
29	20130412063020	20130730090416	3023116	1777213	True	1

In [61]:
```
import pandas as pd
```

In [62]:
```
data = {'user' : user,
        'gender' : gender}
```

In [63]:
```
userngender = pd.DataFrame(data)
```

In [64]:
```
print userngender[10:20]
```

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

```
import dump_svmlight_file
file = open('user_app_predicted.df', 'w')
pickle.dump(uapp, f)
```

In [72]:
```
import pickle
filename="profiled_user_gender.pkl"
fout = file(filename,"w")
pickle.dump(userngender,fout)
```

In [74]:
```
from pandas import DataFrame
```

In [76]:
```
DataFrame.save(userngender, filename)
```

In [ ]:
```

```

In [ ]:
```
uapp = pickle.load(file('user_app_predicted.df'))
```

In [85]:
```
user_id_[14995:15000]
a = uapp.ix[ (uapp['user_id'] == 3051549 ), 'gender'].unique()
a = int(a)
```

Out[85]:
```
array([3780729, 1363769, 1721736, 3051549, 3823028])
```

In [1]:
```
user_gender = []
```

In [ ]:
```
for x in user_id_[:15000]:
    a = uapp.ix[ (uapp['user_id'] == x ), 'gender'].unique()
    a = int(a)
    temp = [x, a]
    user_gender.append(temp)
```

In [98]:
```

```

Out[98]:
```
set([2, 3823028])
```

In [ ]:
```

```