Asana DS challenge by Jia Guo

February 25, 2018

```
In [1]: import pandas as pd
        import numpy as np
        from datetime import timedelta
```

```
In [2]: # import user_engagement by pandas dataframe
        df = pd.read_csv('takehome_user_engagement.csv')
```

```
In [3]: # get the total log-in frequencies of each user regardless of login time
        total_login = df.groupby('user_id').size().reset_index(name='total_login_freqencies')
        total_login.head()
```

```
Out[3]:    user_id  total_login_freqencies
        0        1                       1
        1        2                      14
        2        3                       1
        3        4                       1
        4        5                       1
```

```
In [4]: # drop the rows which the total log-in freqencies are less than 3
        total_login = total_login[total_login.total_login_freqencies >= 3]
        total_login.head()
```

```
Out[4]:     user_id  total_login_freqencies
        1         2                      14
        7        10                     284
        13       20                       7
        24       33                      18
        28       42                     342
```

```
In [5]: # create a merged_inner dataframe
        # to start analyze the time_stamp in each 7-day period to keep eliminating users
        merged_inner = pd.merge(left=df,right=total_login, left_on='user_id', right_on='user_i
        merged_inner.head()
```

```
Out[5]:        time_stamp  user_id  visited  total_login_freqencies
        0  11/15/13 3:45        2        1                      14
        1  11/29/13 3:45        2        1                      14
        2   12/9/13 3:45        2        1                      14
        3  12/25/13 3:45        2        1                      14
        4  12/31/13 3:45        2        1                      14
```

```
In [6]: # delete the time part in the time_stamp
        merged_inner['time_stamp'] = pd.to_datetime(merged_inner['time_stamp'], errors='coerce
        merged_inner['time_stamp'] = merged_inner['time_stamp'].dt.date
        dfnew = merged_inner.drop(['visited', 'total_login_freqencies'], 1)
        dfnew.head()
```

```
Out[6]:    time_stamp  user_id
        0  2013-11-15        2
        1  2013-11-29        2
        2  2013-12-09        2
        3  2013-12-25        2
        4  2013-12-31        2
```

```
In [7]: # check time of each column make sure error-free to calculate the log-in gap-day of ea
        dfnew['time_stamp'] = pd.to_datetime(dfnew['time_stamp'])
        dfnew.dtypes
```

```
Out[7]: time_stamp    datetime64[ns]
        user_id                int64
        dtype: object
```

```
In [8]: # calculate the gap during each 3 times of login for each user
        dfnew['lastlast_login'] = dfnew.groupby('user_id')['time_stamp'].shift(2)
        dfnew['time_diff'] = dfnew['time_stamp'] - dfnew['lastlast_login']
        dfnew.head()
```

```
Out[8]:   time_stamp  user_id lastlast_login time_diff
        0 2013-11-15        2            NaT       NaT
        1 2013-11-29        2            NaT       NaT
        2 2013-12-09        2     2013-11-15   24 days
        3 2013-12-25        2     2013-11-29   26 days
        4 2013-12-31        2     2013-12-09   22 days
```

```
In [9]: # remove all rows with NaT value in time_diff column
        dfnew = dfnew[dfnew.time_diff.notnull()]
        dfnew.head()
```

```
Out[9]:   time_stamp  user_id lastlast_login time_diff
        2 2013-12-09        2     2013-11-15   24 days
        3 2013-12-25        2     2013-11-29   26 days
        4 2013-12-31        2     2013-12-09   22 days
        5 2014-01-08        2     2013-12-25   14 days
        6 2014-02-03        2     2013-12-31   34 days
```

```
In [10]: # convert the time_diff components to integer in days
         dfnew['time_diff'] = (dfnew.time_diff / np.timedelta64(1, 'D')).astype(int)
         dfnew.head()
```

```
Out[10]:   time_stamp  user_id lastlast_login  time_diff
         2 2013-12-09        2     2013-11-15         24
```

```
3 2013-12-25        2     2013-12-09        26
```

```
3 2013-12-25        2     2013-11-29        26
4 2013-12-31        2     2013-12-09        22
5 2014-01-08        2     2013-12-25        14
6 2014-02-03        2     2013-12-31        34
```

```
In [11]: # the user_id showed in below output dataframe should be the adopted user
         dfnew = dfnew.drop(dfnew[dfnew['time_diff']>7].index)
         dfnew = dfnew.drop_duplicates(subset=['user_id'])
         dfnew.head()
```

```
Out[11]:     time_stamp  user_id lastlast_login  time_diff
         8   2014-02-09        2     2014-02-03          6
         18  2013-02-06       10     2013-01-30          7
         300 2014-03-13       20     2014-03-11          2
         308 2014-03-23       33     2014-03-17          6
         327 2012-12-25       42     2012-12-18          7
```

```
In [12]: # get the current existing adopted user list by user id
         adopted_user_list = []
         adopted_user_list = dfnew['user_id'].values
         num_adopted_user = len(adopted_user_list)
         print('number of current existing adopted user are:', num_adopted_user)
         print('and their user ids are:', adopted_user_list)
```

```
number of current existing adopted user are: 1656
and their user ids are: [    2    10    20 ... 11969 11975 11988]
```

```
In [13]: # read in the takehome_users.csv
         df1 = pd.read_csv('takehome_users.csv', encoding='mac_roman')
         df1.head()
```

```
Out[13]:    object_id  creation_time               name                       email  \
         0          1   4/22/14 3:53     Clausen August    AugustCClausen@yahoo.com
         1          2  11/15/13 3:45      Poole Matthew      MatthewPoole@gustr.com
         2          3  3/19/13 23:14  Bottrill Mitchell  MitchellBottrill@gustr.com
         3          4   5/21/13 8:09    Clausen Nicklas   NicklasSClausen@yahoo.com
         4          5  1/17/13 10:14          Raw Grace          GraceRaw@yahoo.com

          creation_source  last_session_creation_time  opted_in_to_mailing_list  \
         0   GUEST_INVITE                1.398139e+09                         1
         1     ORG_INVITE                1.396238e+09                         0
         2     ORG_INVITE                1.363735e+09                         0
         3   GUEST_INVITE                1.369210e+09                         0
         4   GUEST_INVITE                1.358850e+09                         0

            enabled_for_marketing_drip  org_id  invited_by_user_id email_domain
         0                           0      11             10803.0    yahoo.com
         1                           0       1               316.0    gustr.com
```

```
2                          0     94             1525.0    gustr.com
3                          0      1             5151.0    yahoo.com
4                          0    193             5240.0    yahoo.com
```

```
In [14]: # takes only some specific columns
         df1 = df1.drop(['creation_time', 'name', 'email' ], axis=1)
         df1.head()
```

```
Out[14]:    object_id creation_source  last_session_creation_time  \
         0          1    GUEST_INVITE                1.398139e+09
         1          2      ORG_INVITE                1.396238e+09
         2          3      ORG_INVITE                1.363735e+09
         3          4    GUEST_INVITE                1.369210e+09
         4          5    GUEST_INVITE                1.358850e+09

            opted_in_to_mailing_list  enabled_for_marketing_drip  org_id  \
         0                         1                           0      11
         1                         0                           0       1
         2                         0                           0      94
         3                         0                           0       1
         4                         0                           0     193

            invited_by_user_id email_domain
         0             10803.0    yahoo.com
         1               316.0    gustr.com
         2              1525.0    gustr.com
         3              5151.0    yahoo.com
         4              5240.0    yahoo.com
```

```
In [15]: # convert the unix timestamp to readable datetime
         df1['last_session_creation_time'] = (pd.to_datetime(df1['last_session_creation_time']
```

```
In [16]: # merge two dataframe to discover the hidden pattern in the current exiting adopted u
         dfnew1 = pd.merge(left=df1,right=dfnew, left_on='object_id', right_on='user_id')
         dfnew1 = dfnew1.drop(['time_stamp', 'user_id', 'lastlast_login', 'time_diff', ], axis
         dfnew1.head(10)
```

```
Out[16]:    object_id     creation_source last_session_creation_time  \
         0          2          ORG_INVITE        2014-03-31 03:45:04
         1         10          ORG_INVITE        2014-06-03 22:08:03
         2         20              SIGNUP        2014-05-29 11:46:38
         3         33        GUEST_INVITE        2014-05-31 06:29:09
         4         42              SIGNUP        2014-05-25 19:05:07
         5         43        GUEST_INVITE        2013-04-15 07:13:17
         6         50        GUEST_INVITE        2012-10-23 11:02:08
         7         53        GUEST_INVITE        2013-05-05 23:47:15
         8         60          ORG_INVITE        2014-05-15 22:56:03
         9         63  SIGNUP_GOOGLE_AUTH        2014-06-04 16:30:52
```

```
   opted_in_to_mailing_list  enabled_for_marketing_drip  org_id  \
0                         0                           0       1   
1                         1                           1     318   
2                         0                           0      58   
3                         0                           0     401   
4                         1                           0     235   
5                         0                           0      63   
6                         0                           0      61   
7                         0                           0      37   
8                         0                           0      88   
9                         0                           0     203   

   invited_by_user_id email_domain  
0               316.0    gustr.com  
1              4143.0    gustr.com  
2                 NaN    uhzdq.com  
3                79.0     cuvox.de  
4                 NaN     cuvox.de  
5               149.0    yyyxt.com  
6                50.0    gmail.com  
7              3641.0    gmail.com  
8              3463.0    gmail.com  
9                 NaN    gmail.com  
```

In [17]:
```
# analyzing the email domain
temp1 = dfnew1.groupby(['email_domain']).size().reset_index(name='count')
temp1.sort_values('count', inplace=True)
temp1['email_domain_percent'] = temp1['count']*100/num_adopted_user
temp1.sort_values('email_domain_percent', ascending=False, inplace=True)
temp1.head(10)
```

Out[17]:
```
         email_domain  count  email_domain_percent
43          gmail.com    557             33.635266
159         yahoo.com    267             16.123188
50        hotmail.com    205             12.379227
64     jourrapide.com    170             10.265700
46          gustr.com    150              9.057971
19           cuvox.de    144              8.695652
9           bztuu.com      1              0.060386
2           aosyq.com      1              0.060386
3           bawmq.com      1              0.060386
4           bgdtm.com      1              0.060386
```

In [18]:
```
# analyzing the creation source
temp = dfnew1.groupby(['creation_source']).size().reset_index(name='count')
temp['source_percent'] = temp['count']*100/num_adopted_user
temp.sort_values('source_percent', ascending=False, inplace=True)
temp
```

```
Out[18]:       creation_source  count  source_percent
         1          ORG_INVITE    574       34.661836
         0        GUEST_INVITE    369       22.282609
         3              SIGNUP    302       18.236715
         4  SIGNUP_GOOGLE_AUTH    239       14.432367
         2   PERSONAL_PROJECTS    172       10.386473
```

```
In [19]: # calculate the percentage of opted_in_to_mailing_list of all existing adopted users
         total = sum(df1['opted_in_to_mailing_list'])
         percent_optin_mailinglist = total/num_adopted_user
         print ('About', int(percent_optin_mailinglist),'percent of the current adopted users c

About 1 percent of the current adopted users choose to opt-in the mailing list
```

```
In [20]: # calculate the percentage of enabled_for_marketing_drip of all existing adopted users
         total1 = sum(df1['enabled_for_marketing_drip'])
         percent_enable_marketing_drip = total1/num_adopted_user
         print ('About', int(percent_enable_marketing_drip),'percent of the current adopted use

About 1 percent of the current adopted users choose to opt-in the mailing list
```

```
In [21]: # filter out the top 10 organization ID
         temp2 = dfnew1
         temp2 = temp2[temp2.creation_source.str.contains("ORG_INVITE") == True]
         temp2 = temp2.groupby(['org_id']).size().reset_index(name='count')
         temp2.sort_values('count', inplace=True, ascending=False)
         temp2.head(10)
```

```
Out[21]:      org_id  count
         9         9      8
         3         3      6
         6         6      6
         58       61      5
         95      106      5
         20       20      5
         185     240      5
         49       52      5
         1         1      5
         8         8      5
```

```
In [22]: # filter out the top 10 existing users who invited others to use the product
         temp3 = dfnew1
         temp3 = temp3[temp3.creation_source.str.contains("GUEST_INVITE") == True]
         temp3 = temp3.groupby(['invited_by_user_id']).size().reset_index(name='count')
         temp3.sort_values('count', inplace=True, ascending=False)
         temp3.head(10)
```

```
Out[22]:      invited_by_user_id  count
         211              7107.0      3
         137              4762.0      2
         323             11297.0      2
         304             10628.0      2
         303             10624.0      2
         75               2771.0      2
         76               2776.0      2
         286              9726.0      2
         142              4908.0      2
         110              3819.0      2
```

```
In [23]: # get the detail count of last session creation time by specific date
         temp4 = dfnew1
         temp4['last_session_creation_time'] = temp4['last_session_creation_time'].dt.date
         temp4 = temp4.groupby(['last_session_creation_time']).size().reset_index(name='count'
         temp4.sort_values('count', inplace=True, ascending=False)
         temp4
```

```
Out[23]:     last_session_creation_time  count
         331                 2014-06-04    353
         330                 2014-06-03    136
         329                 2014-06-02     89
         328                 2014-06-01     65
         326                 2014-05-30     65
         323                 2014-05-27     60
         327                 2014-05-31     49
         321                 2014-05-25     49
         320                 2014-05-24     43
         325                 2014-05-29     43
         317                 2014-05-21     43
         322                 2014-05-26     43
         324                 2014-05-28     42
         318                 2014-05-22     38
         319                 2014-05-23     36
         314                 2014-05-18     15
         316                 2014-05-20     14
         315                 2014-05-19     13
         311                 2014-05-15      8
         309                 2014-05-13      8
         301                 2014-05-05      8
         308                 2014-05-12      7
         313                 2014-05-17      6
         310                 2014-05-14      5
         306                 2014-05-10      5
         300                 2014-05-04      5
         312                 2014-05-16      4
         307                 2014-05-11      4
```

```
203                 2013-11-15      4
286                 2014-04-15      3
..                         ...    ...
215                 2013-12-12      1
69                  2013-02-16      1
187                 2013-10-11      1
186                 2013-10-08      1
170                 2013-09-07      1
159                 2013-08-16      1
160                 2013-08-19      1
161                 2013-08-20      1
162                 2013-08-22      1
75                  2013-02-26      1
164                 2013-08-25      1
165                 2013-09-01      1
1                   2012-07-06      1
167                 2013-09-04      1
168                 2013-09-05      1
169                 2013-09-06      1
172                 2013-09-10      1
185                 2013-10-07      1
73                  2013-02-24      1
72                  2013-02-23      1
175                 2013-09-13      1
176                 2013-09-14      1
71                  2013-02-22      1
178                 2013-09-17      1
179                 2013-09-21      1
180                 2013-09-26      1
181                 2013-09-27      1
183                 2013-10-01      1
184                 2013-10-05      1
0                   2012-07-02      1

[332 rows x 2 columns]
```