

Yammer Engagement

December 20, 2017

1 Engagement Dashboard

```
In [1]: '''
        Import dataframes that contains all
        records returned by queries
        '''

        import pandas as pd

        events_q = pd.read_csv('events_q.csv')
        signups = pd.read_csv('signups.csv')
        retention = pd.read_csv('Retention_Rate.csv')
        device = pd.read_csv('Device.csv')
        etypes = pd.read_csv('Email_Types.csv')
        erates = pd.read_csv('Email_Rates.csv')

SELECT DATE_TRUNC('week', e.occurred_at),
        COUNT(DISTINCT e.user_id) AS weekly_active_users
FROM yammer_events e
WHERE e.event_type = 'engagement'
AND e.event_name = 'login'
GROUP BY 1
ORDER BY 1
```

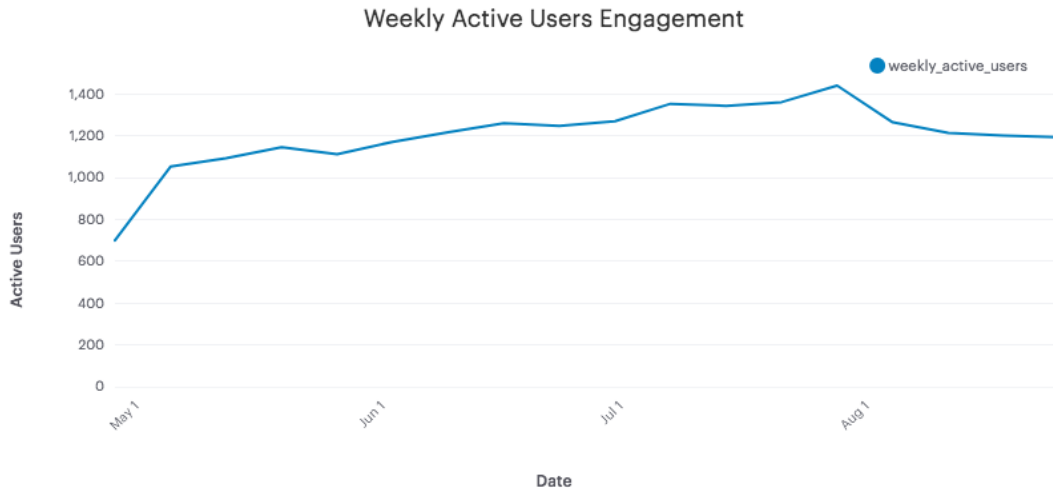
```
In [2]: events_q.head()
```

```
Out [2]:
```

	date_trunc	weekly_active_users
0	2014-04-28 00:00:00	701
1	2014-05-05 00:00:00	1054
2	2014-05-12 00:00:00	1094
3	2014-05-19 00:00:00	1147
4	2014-05-26 00:00:00	1113

```
In [3]: from IPython.core.display import Image
        Image(filename="Engagement_Dashboard.png")
```

```
Out [3]:
```



2 Daily Signups

```
SELECT DATE_TRUNC('day', u.created_at) AS "Day",
       COUNT(*) as "As users",
       COUNT(CASE WHEN u.activated_at IS NOT NULL then u.user_id ELSE NULL END) AS
         "Activated Users"
FROM yammer_users u
WHERE created_at >= '2014-06-01' AND created_at <= '2014-09-01'
GROUP BY "Day"
ORDER BY "Day"
```

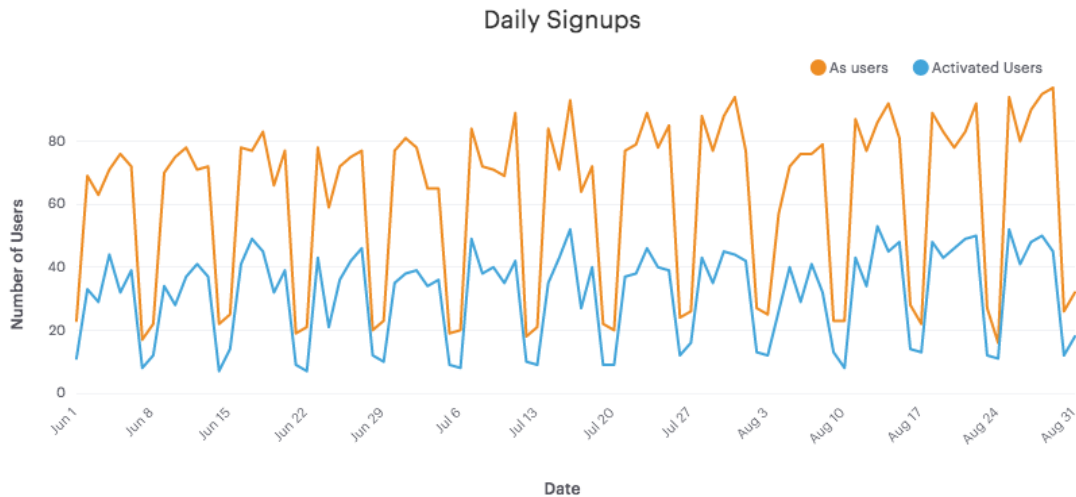
```
In [4]: signups.head()
```

```
Out [4]:
```

	Day	As users	Activated Users
0	2014-06-01 00:00:00	23	11
1	2014-06-02 00:00:00	69	33
2	2014-06-03 00:00:00	63	29
3	2014-06-04 00:00:00	71	44
4	2014-06-05 00:00:00	76	32

```
In [5]: from IPython.core.display import Image
        Image(filename=("Daily_Signups_Graph.png"))
```

```
Out [5]:
```



3 Retention Rate

```

SELECT DATE_TRUNC('week', z.occurred_at) AS "Week",
       AVG(z.event_account_age) AS "Average Age During Week",
       COUNT(DISTINCT CASE WHEN z.user_age > 70 THEN z.user_id ELSE NULL END) AS
         "10+ Weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 70 AND z.user_age >= 63 THEN
         z.user_id ELSE NULL END) AS "9 Weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 63 AND z.user_age >= 56 THEN
         z.user_id ELSE NULL END) AS "8 Weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 56 AND z.user_age >= 49 THEN
         z.user_id ELSE NULL END) AS "7 Weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 49 AND z.user_age >= 42 THEN
         z.user_id ELSE NULL END) AS "6 Weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 42 AND z.user_age >= 35 THEN
         z.user_id ELSE NULL END) AS "5 Weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 35 AND z.user_age >= 28 THEN
         z.user_id ELSE NULL END) AS "4 Weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 28 AND z.user_age >= 21 THEN
         z.user_id ELSE NULL END) AS "3 Weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 21 AND z.user_age >= 14 THEN
         z.user_id ELSE NULL END) AS "2 Weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 14 AND z.user_age >= 7 THEN
         z.user_id ELSE NULL END) AS "1 Week",
       COUNT(DISTINCT CASE WHEN z.user_age < 7 THEN z.user_id ELSE NULL END)
         AS "Less than a week"

FROM (
  SELECT e.occurred_at,

```

```

        u.user_id,
        DATE_TRUNC('week', u.activated_at) AS "activation_week",
        EXTRACT('day' FROM e.occurred_at - u.activated_at)
            AS "event_account_age",
        EXTRACT('day' FROM '2014-09-01'::TIMESTAMP - u.activated_at)
            AS "user_age"
    FROM yammer_users u
    JOIN yammer_events e
    ON e.user_id = u.user_id
    AND e.event_type = 'engagement'
    AND e.event_name = 'login'
    AND e.occurred_at >= '2014-05-01'
    AND e.occurred_at < '2014-09-01'
    WHERE u.activated_at IS NOT NULL
) z
GROUP BY "Week"
ORDER BY "Week"
LIMIT 100

```

In [6]: retention.head()

```

Out [6]:
           Week  Average Age During Week  10+ Weeks  9 Weeks  8 Weeks
0  2014-04-28 00:00:00                124.007239         701         0
1  2014-05-05 00:00:00                124.381691        1054         0
2  2014-05-12 00:00:00                131.938644        1094         0
3  2014-05-19 00:00:00                132.326628        1147         0
4  2014-05-26 00:00:00                132.345363        1113         0

           7 Weeks  6 Weeks  5 Weeks  4 Weeks  3 Weeks  2 Weeks  1 Week  \
0             0         0         0         0         0         0         0
1             0         0         0         0         0         0         0
2             0         0         0         0         0         0         0
3             0         0         0         0         0         0         0
4             0         0         0         0         0         0         0

           Less than a week
0             0
1             0
2             0
3             0
4             0

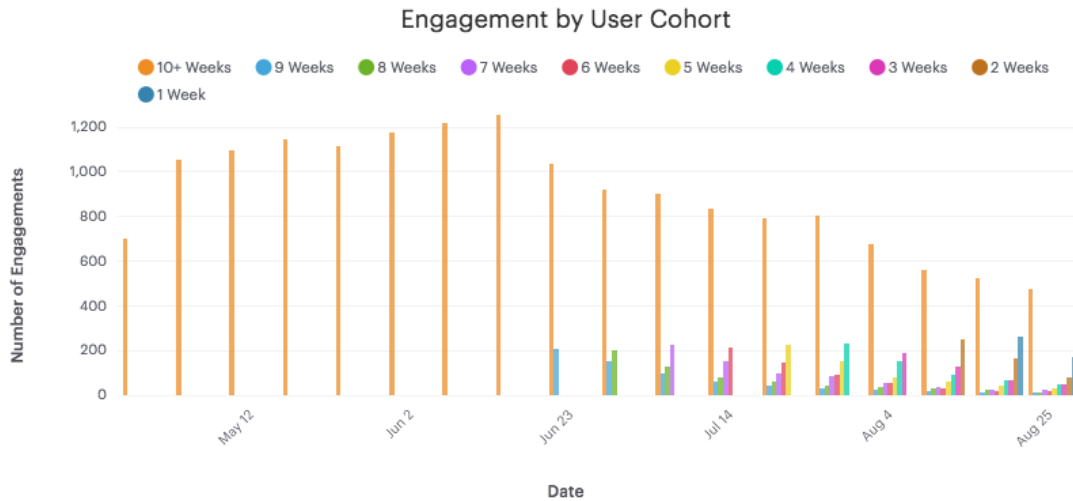
```

```

In [7]: from IPython.core.display import Image
        Image(filename=("Engagement_by_User_Cohort.png"))

```

Out [7]:



4 Device Type

```
SELECT DATE_TRUNC('week', e.occurred_at) AS "Week",
       COUNT(DISTINCT e.user_id) AS weekly_active_users,
       COUNT(DISTINCT CASE WHEN e.device IN ('macbook pro','acer aspire notebook',
       'acer aspire desktop','lenovo thinkpad',
       'mac mini','dell inspiron notebook',
       'dell inspiron desktop','macbook air',
       'asus chromebook','hp pavilion desktop')
       THEN e.user_id ELSE NULL END) AS "Computer",
       COUNT(DISTINCT CASE WHEN e.device IN ('iphone 5s','samsung galaxy note',
       'nokia lumia 635','amazon fire phone',
       'nexus 5','iphone 4s',
       'htc one','iphone 5',
       'samsung galaxy s4') THEN e.user_id ELSE NULL END)
       AS "Phone",
       COUNT(DISTINCT CASE WHEN e.device IN ('kindle fire','ipad mini',
       'samsung galaxy tablet','windows surface','ipad air','nexus 7',
       'nexus 10') THEN e.user_id ELSE NULL END) AS "Tablet"
FROM yammer_events e
WHERE e.event_type = 'engagement'
AND e.event_name = 'login'
GROUP BY "Week"
ORDER BY "Week"
```

In [8]: device.head()

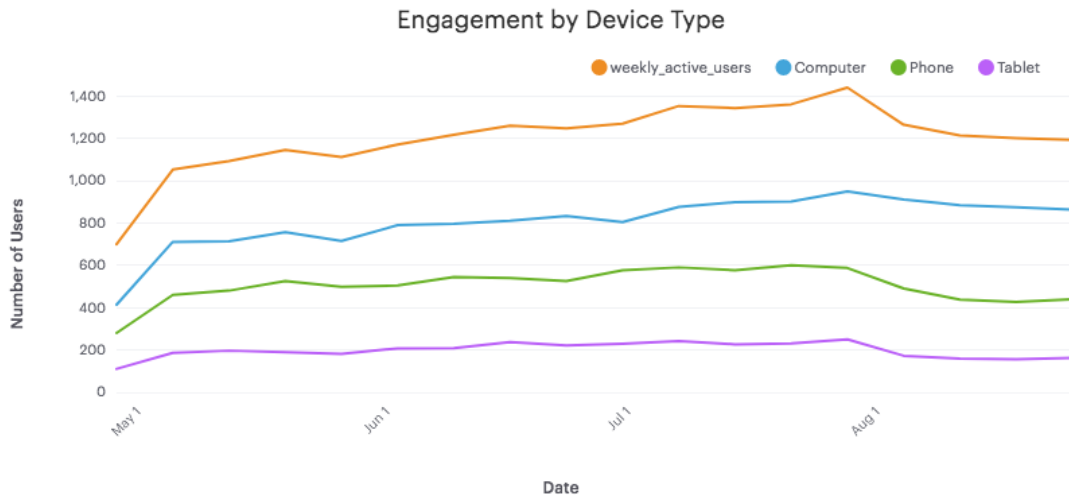
```
Out [8]:
```

	Week	weekly_active_users	Computer	Phone	Tablet
0	2014-04-28 00:00:00	701	415	281	111

1	2014-05-05 00:00:00	1054	712	461	187
2	2014-05-12 00:00:00	1094	715	481	197
3	2014-05-19 00:00:00	1147	758	526	190
4	2014-05-26 00:00:00	1113	716	500	182

```
In [9]: from IPython.core.display import Image
Image(filename="Engagement_by_Device_Type.png")
```

Out [9]:



5 Email Types

```
SELECT DATE_TRUNC('week', e.occurred_at) AS "Week",
COUNT(CASE WHEN e.action = 'sent_weekly_digest' THEN e.user_id
ELSE NULL END) AS "Weekly Emails",
COUNT(CASE WHEN e.action = 'sent_reengagement_email' THEN e.user_id
ELSE NULL END) AS "Reengagement Emails",
COUNT(CASE WHEN e.action = 'email_open' THEN e.user_id
ELSE NULL END) AS "Open Emails",
COUNT(CASE WHEN e.action = 'email_clickthrough' THEN e.user_id
ELSE NULL END) AS "Email Clickthroughs"
FROM yammer_emails e
GROUP BY "Week"
ORDER BY "Week"
```

```
In [10]: etypes.head()
```

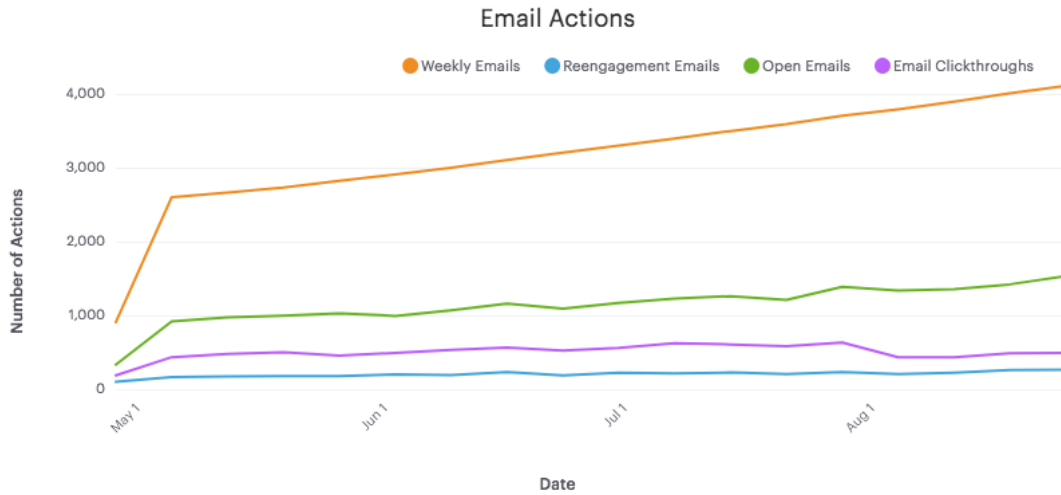
Out [10]:	week	Weekly Emails	Reengagement Emails	Open Emails
0	2014-04-28 00:00:00	908	98	332
1	2014-05-05 00:00:00	2602	164	919

2	2014-05-12 00:00:00	2665	175	971
3	2014-05-19 00:00:00	2733	179	995
4	2014-05-26 00:00:00	2822	179	1026

Email Clickthroughs	
0	187
1	434
2	479
3	498
4	453

```
In [11]: from IPython.core.display import Image
Image(filename="Number_of_Email_Actions.png")
```

Out[11]:



6 Email Rates

```
SELECT week,
       weekly_opens/CASE WHEN weekly_emails = 0 THEN 1 ELSE weekly_emails
       END::FLOAT AS weekly_open_rate,
       weekly_ctr/CASE WHEN weekly_opens = 0 THEN 1 ELSE weekly_opens
       END::FLOAT AS weekly_ctr,
       retain_opens/CASE WHEN retain_emails = 0 THEN 1 ELSE retain_emails
       END::FLOAT AS retain_open_rate,
       retain_ctr/CASE WHEN retain_opens = 0 THEN 1 ELSE retain_opens
       END::FLOAT AS retain_ctr
FROM(
  SELECT DATE_TRUNC('week',e1.occurred_at) AS week,
         COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e1.user_id
```

```

        ELSE NULL END) AS weekly_emails,
COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e2.user_id
        ELSE NULL END) AS weekly_opens,
COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e3.user_id
        ELSE NULL END) AS weekly_ctr,
COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e1.user_id
        ELSE NULL END) AS retain_emails,
COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e2.user_id
        ELSE NULL END) AS retain_opens,
COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e3.user_id
        ELSE NULL END) AS retain_ctr
FROM yammer_emails e1
LEFT JOIN yammer_emails e2
  ON e2.occurred_at >= e1.occurred_at
  AND e2.occurred_at < e1.occurred_at + INTERVAL '5 MINUTE'
  AND e2.user_id = e1.user_id
  AND e2.action = 'email_open'
LEFT JOIN tutorial.yammer_emails e3
  ON e3.occurred_at >= e2.occurred_at
  AND e3.occurred_at < e2.occurred_at + INTERVAL '5 MINUTE'
  AND e3.user_id = e2.user_id
  AND e3.action = 'email_clickthrough'
WHERE e1.occurred_at >= '2014-06-01'
AND e1.occurred_at < '2014-09-01'
AND e1.action IN ('sent_weekly_digest','sent_reengagement_email')
GROUP BY week
) a
ORDER BY week

```

In [12]: erates.head()

```

Out[12]:
           week  weekly_open_rate  weekly_ctr  retain_open_rate  \
0  2014-05-26 00:00:00      0.000000      0.000000      0.837838
1  2014-06-02 00:00:00      0.281690      0.417073      0.869347
2  2014-06-09 00:00:00      0.299034      0.419822      0.905263
3  2014-06-16 00:00:00      0.306280      0.394322      0.897436
4  2014-06-23 00:00:00      0.289055      0.401294      0.871658

           retain_ctr
0      0.903226
1      0.867052
2      0.906977
3      0.895238
4      0.932515

```

In [13]: `from IPython.core.display import Image`
`Image(filename="Email_Rates.png")`

Out[13]:

Email Open and CTR Rates

