Yammer

Yammer is a social network for communicating with coworkers. Individuals share documents, updates, and ideas by posting them in groups. Yammer is free to use indefinitely, but companies must pay license fees if they want access to administrative controls, including integration with user management systems like ActiveDirectory.

Yammer has a centralized Analytics team, which sits in the Engineering organization. Their primary goal is to drive better product and business decisions using data. They do this partially by providing tools and education that make other teams within Yammer more effective at using data to make better decisions. They also perform ad-hoc analysis to support specific decisions.

The Yammer analytics philosophy

Yammer analysts are trained to constantly consider the value of each individual project; they seek to maximize the return on their time. Analysts typically opt for less precise solutions to problems if it means investing substantially less time as well.

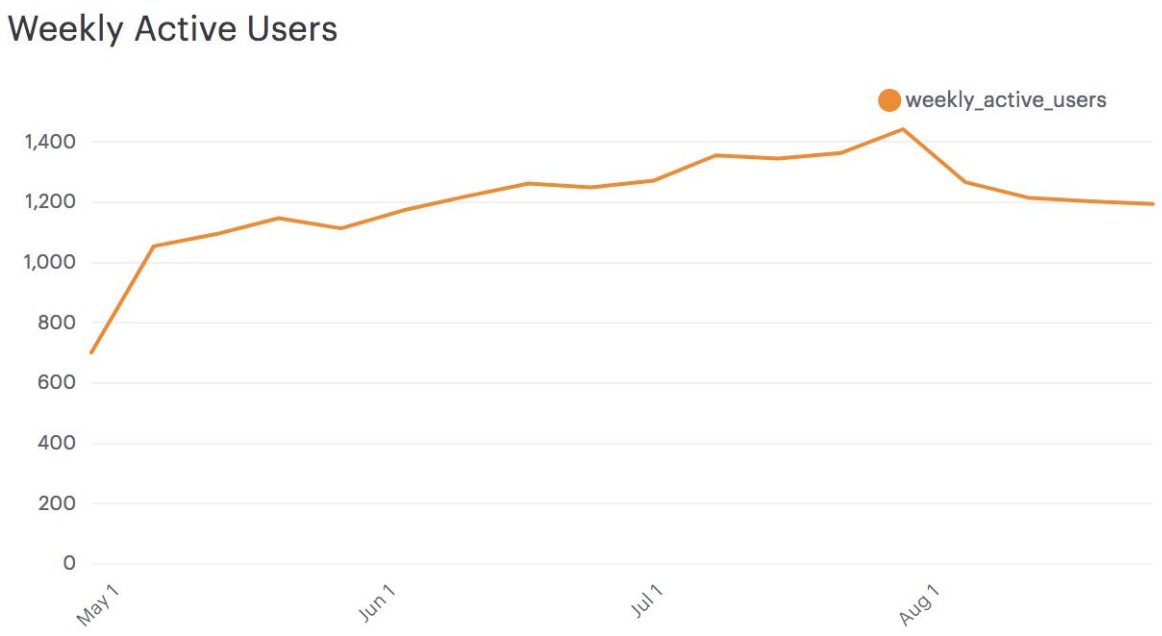
They are also taught to consider the impact of everything on the company at large. This includes high-level decision making like choosing which projects to prioritize. It also influences the way analysts think about metrics. Product decisions are always evaluated against core engagement, retention, and growth metrics in addition to product-specific usage metrics (like, for example, the number of times someone views another user's profile).

Investigating a Drop in User Engagement

Yammer's Analysts are responsible for triaging product and business problems as they come up. In many cases, these problems surface through key metric dashboards that execs and managers check daily.

The problem

You show up to work Tuesday morning, September 2, 2014. The head of the Product team walks over to your desk and asks you what you think about the latest activity on the user engagement dashboards. You fire them up, and something immediately jumps out:



Interactive Engagement Dashboard:
https://modeanalytics.com/modeanalytics/reports/cbb8c291ee96/runs/7925c979521e/viz/cfcdb6b78885

The above chart shows the number of engaged users each week. Yammer defines engagement as having made some type of server call by interacting with the product (shown in the data as events of type “engagement”). Any point in this chart can be interpreted as “the number of users who logged at least one engagement event during the week starting on that date.”

df.loc[:, df.isnull().any()]

Getting oriented

Before you even touch the data, come up with a list of possible causes for the dip in retention shown in the chart above. Make a list and determine the order in which you will check them. Make sure to note how you will test each hypothesis. Think carefully about the criteria you use to order them and write down the criteria as well.

Also, make sure you understand what the above chart shows and does not show.

Digging in

Once you have an ordered list of possible problems, it’s time to investigate.

For this problem, you will need to use four tables. The tables names and column definitions are listed below—click a table name to view information about that table. Note: this data is fake and was generated for the purpose of this case study. It is similar in structure to Yammer's actual data, but for privacy and security reasons it is not real.

Table 1: Users

yammer_useruserss.csv

This table includes one row per user, with descriptive information about that user's account.

user_id	A unique ID per user. Can be joined to user_id in either of the other tables.
created_at	The time the user was created (first signed up)
state	The state of the user (active or pending)
activated_at	The time the user was activated, if they are active
company_id	The ID of the user's company
language	The chosen language of the user

Table 2: Events

yammer_events.csv

This table includes one row per event, where an event is an action that a user has taken on Yammer. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.

user_id	The ID of the user logging the event. Can be joined to user_id in either of the other tables.
occurred_at	The time the event occurred.
event_type	The general event type. There are two values in this dataset: "signup_flow", which refers to anything occurring during the process of a user's authentication, and "engagement", which refers to general product usage after the user has signed up for the first time.
event_name	The specific action the user took. Possible values include: • **create_user:** User is added to Yammer's database during signup process • **enter_email:** User begins the signup process by entering her email address • **enter_info:** User enters her name and personal information during signup process • **complete_signup:** User completes the entire signup/authentication process • **home_page:** User loads the home page • **like_message:** User likes another user's message • **login:** User logs into Yammer • **search_autocomplete:** User selects a search result from the

	autocomplete list • **search_run:** User runs a search query and is taken to the search results page • **search_click_result_X:** User clicks search result X on the results page, where X is a number from 1 through 10. • **send_message:** User posts a message • **view_inbox:** User views messages in her inbox
location	The country from which the event was logged (collected through IP address).
device:	The type of device used to log the event.

Table 3: Email Events

yammer_emails.csv

This table contains events specific to the sending of emails. It is similar in structure to the events table above.

user_id	The ID of the user to whom the event relates. Can be joined to user_id in either of the other tables.
occurred_at	The time the event occurred.
action	The name of the event that occurred. "sent_weekly_digest" means that the user was delivered a digest email showing relevant conversations from the previous day. "email_open" means that the user opened the email. "email_clickthrough" means that the user clicked a link in the email.

Table 4: Rollup Periods

dimension_rollup_periods.csv

The final table is a lookup table that is used to create rolling time periods. Though you could use the INTERVAL() function, creating rolling time periods is often easiest with a table like this. You won't necessarily need to use this table in queries that you write, but the column descriptions are provided here so that you can understand the query that creates the chart shown above.

period_id	This identifies the type of rollup period. The above dashboard uses period 1007, which is rolling 7-day periods.
time_id	This is the identifier for any given data point — it's what you would put on a chart axis. If time_id is 2014-08-01, that means that is represents the rolling 7-day period leading up to 2014-08-01.
pst_start	The start time of the period in PST. For 2014-08-01, you'll notice that this is 2014-07-25 — one week prior. Use this to join events to the table.
pst_end	The start time of the period in PST. For 2014-08-01, the end time is 2014-08-01.
utc_start	The same as pst_start, but in UTC time.
pst_start	The same as pst_end, but in UTC time.

Making a recommendation

Start to work your way through your list of hypotheses in order to determine the source of the drop in engagement.

Answer the following questions:

- Do the answers to any of your original hypotheses lead you to further questions?
- If so, what are they and how will you test them?
- If they are questions that you can't answer using data alone, how would you go about answering them (hypothetically, assuming you actually worked at this company)?
- What seems like the most likely cause of the engagement dip?
- What, if anything, should the company do in response?