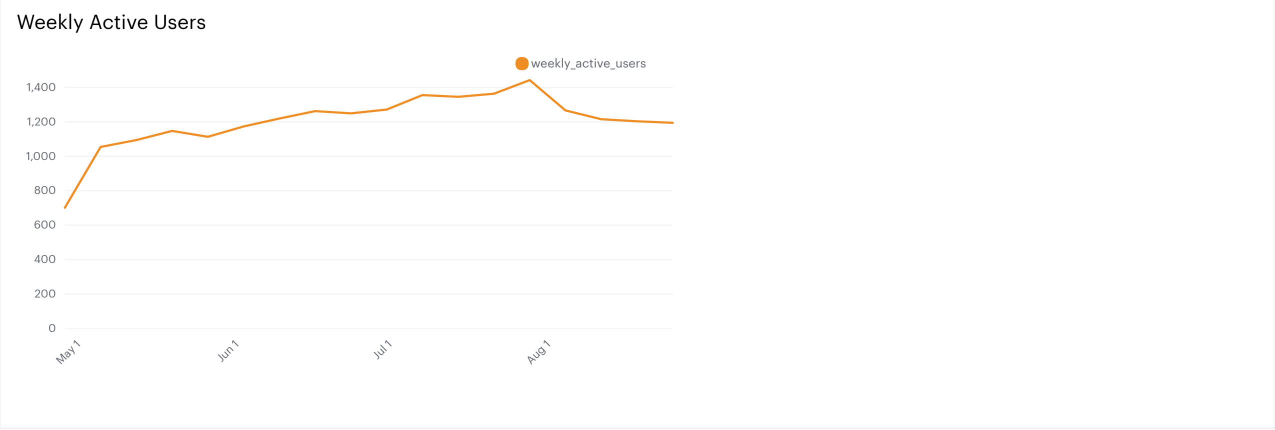
Investigating a Drop in User Engagement

Before starting, be sure to read the overview to learn a bit about Yammer as a company.

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:



(https://modeanalytics.com/) Powered by Mode (https://modeanalytics.com/modeanalytics/reports/cbb8c291ee96/runs/7925c979521e)

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.”

You are responsible for determining what caused the dip at the end of the chart shown above and, if appropriate, recommending solutions for the problem.

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.

If you want to check your list of possible causes against ours, read the first part of the answer key.

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

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

This table name in Mode is tutorial.yammer_users

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

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.

This table name in Mode is tutorial.yammer_events

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 occuring 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

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

This table name in Mode is tutorial.yammer_emails

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

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.

This table name in Mode is benn.dimension_rollup_periods

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. You can see how this is used in conjunction with pst_start to join events to this table in the query that produces the above chart.

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. As you explore, make sure to save your work. It may be helpful to start with the code that produces the above query, which you can find by clicking the link in the footer of the chart and navigating to the “query” tab.

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?

Answers

If you want to check your work against the answer key, click here.