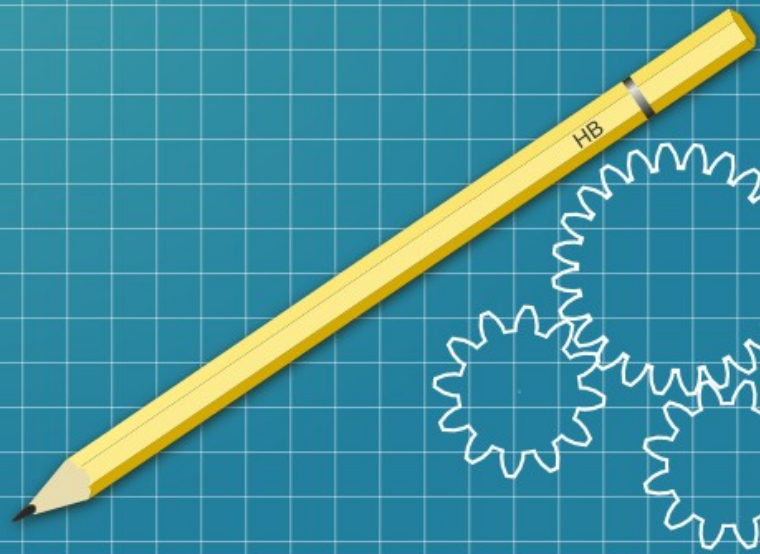
Operation Analytics and Investigating Metric Spike

PROJECT NO.3

NAME :- AYUSH RAVINDRA VAIDANDE



Description:

- Case Study 1 (Job Data) :

  A. Number of jobs reviewed: Amount of jobs reviewed over time.

  Your task: Calculate the number of jobs reviewed per hour per day for November 2020?

  B. Throughput: It is the no. of events happening per second.

  Your task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

  C. Percentage share of each language: Share of each language for different contents.

  Your task: Calculate the percentage share of each language in the last 30 days?

  D. Duplicate rows: Rows that have the same value present in them.

  Your task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

- Case Study 2 (Investigating metric spike) :

  A. User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service.

  Your task: Calculate the weekly user engagement?

  B. User Growth: Amount of users growing over time for a product.

  Your task: Calculate the user growth for product?

  C. Weekly Retention: Users getting retained weekly after signing-up for a product.

  Your task: Calculate the weekly retention of users-sign up cohort?

  D. Weekly Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.

  Your task: Calculate the weekly engagement per device?

  E. Email Engagement: Users engaging with the email service.

  Your task: Calculate the email engagement metrics?

SOFTWARE USED

- MYSQL WORKBENCH 8.0 CE (COMMUNITY EDITION)

  VERSION :- 8.0

- Number of jobs reviewed: Amount of jobs reviewed over time.Calculate the number of jobs reviewed per hour per day for November 2020?

  To find Number of jobs reviewed reviewed per hour per day for november 2020

- QUERY :-

  SELECT ds AS Dates, ROUND((COUNT(job_id)/sum(time_spent))*3600) AS "jobs reviewed per hour per day" FROM job_data WHERE ds BETWEEN "2020-11-01" AND "2020-11-30" GROUP BY ds;

- OUTPUT :-

Dates	jobs reviewed per hour per day
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35
2020-11-26	64
2020-11-25	80

- Throughput: It is the no. of events happening per second.

  Let’s say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

- Here if we use weekly thoughput :-
- QUERY :-
- SELECT ROUND(COUNT(event)/SUM(time_spent),2) AS "Weekly throughput" FROM job_data
- OUTPUT :-

Weekly throughput
0.03

- But if we use daily throught put
- QUERY :-
- SELECT ds AS Dates, ROUND(COUNT(event)/SUM(time_spent),2) AS "Daily throughput" FROM job_data GROUP by ds ORDER by ds
- ORDER :-

Dates	Daily throughput
2020-11-25	0.02
2020-11-26	0.02
2020-11-27	0.01
2020-11-28	0.06
2020-11-29	0.05
2020-11-30	0.05

- CONCLUSION:-
- Metrics will go up and down on weekly or daily basis, we will get number faster every day or minute if we want, as a result rolling metrics are superb at showing if your metrics are trending up or down on a daily level.

- Percentage share of each language: Share of each language for different contents. Calculate the percentage share of each language in the last 30 days?
- We will first divide the total number of languages (distinct/non-distinct) by the total number of rows presents in the table
- Then we will do the grouping based on the languages
- QUERY :-
- select job_data.job_id, job_data.language, count(distinct job_data.language) as total_of_each_language, ((count(job_data.language)/(select count(*) from job_data))*100) as percentage_share_of_each_distinct_language from job_data group by job_data.language;

- OUTPUT :-

job_id	language	total_of_each_language	percentage_share_of_each_distinct_language
22	Arabic	1	12.5000
21	English	1	12.5000
11	French	1	12.5000
25	Hindi	1	12.5000
20	Italian	1	12.5000
23	Persian	1	37.5000

- Duplicate rows: Rows that have the same value present in them. Let's say you see some duplicate rows in the data. How will you display duplicates from the table?
- QUERY :-
- SELECT actor_id ,COUNT(*) AS Duplicates FROM job_data GROUP by actor_id HAVING COUNT(*) > 1;
- OUTPUT :-

actor_id	Duplicates
1003	2

Case Study 2 (Investigating metric spike)

- User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service.Calculate the weekly user engagement?
- We will extract the week from the occurred_at column of the events table

  using the EXTRACT function and WEEK function
- Then we will be counting the number of distinct user_id from the events table
- Then we will use the GROUP BY function to group the output w.r.t week from occurred_at
- QUERY :-
- SELECT extract (week from occurred_at) as week_number, count(distinct user_id) as number_of_users FROM tutorial.yammer_events group by week_number;

- OUTPUT :-

week_number	number_of_users
18	791
19	1244
20	1270
21	1341
22	1293
23	1366
24	1434
25	1462
26	1443
27	1477
28	1556
29	1556
30	1593
31	1685
32	1483
33	1438
34	1412
35	1442

- User Growth: Amount of users growing over time for a product. Calculate the user growth for product?
- To find the user growth (number of active users per week):-
- First we will the extract the year and week for the occurred_at column of the users table using the extract, year and week functions
- Then we will group the extracted week and year on the basis of year and week number
- Then we ordered the result on the basis of year and week number
- Then we will find the cumm_active_users using the SUM, OVER and ROW function between unbounded preceding and current row

- QUERY :-
- select year_num, week_num, num_active_users, SUM(num_active_users)OVER(ORDER BY year_num, week_num ROWS BETWEEN
- UNBOUNDED PRECEDING AND CURRENT ROW) AS cum_active_users
- from
- (select
- extract (year from a.activated_at) as year_num,
- extract (week from a.activated_at) as week_num,
- count(distinct user_id) as num_active_users
- from
- tutorial.yammer_users a
- WHERE state = 'active' group by year_num,week_num order by year_num,week_num) a;

- OUTPUT :-

year_num	week_num	num_active_users	cum_active_users	year_num	week_num	num_active_users	cum_active_users
2013	1	67	67	2013	45	97	2564
2013	2	29	96	2013	46	94	2658
2013	3	47	143	2013	47	82	2740
2013	4	36	179	2013	48	103	2843
2013	5	30	209	2013	49	96	2939
2013	6	48	257	2013	50	117	3056
2013	7	41	298	2013	51	123	3179
2013	8	39	337	2013	52	104	3283
2013	9	33	370	2014	1	91	3374
2013	10	43	413	2014	2	122	3496
2013	11	33	446	2014	3	112	3608
2013	12	32	478	2014	4	113	3721
2013	13	33	511	2014	5	130	3851
2013	14	40	551	2014	6	132	3983
2013	15	35	586	2014	7	135	4118
2013	16	42	628	2014	8	127	4245
2013	17	48	676	2014	9	127	4372
2013	18	48	724	2014	10	135	4507
2013	19	45	769	2014	11	152	4659
2013	20	55	824	2014	12	132	4791
2013	21	41	865	2014	13	151	4942
2013	22	49	914	2014	14	161	5103
2013	23	51	965	2014	15	166	5269
2013	24	51	1016	2014	16	165	5434
2013	25	46	1062	2014	17	176	5610
2013	26	57	1119	2014	18	172	5782
2013	27	57	1176	2014	19	160	5942
2013	28	52	1228	2014	20	186	6128
2013	29	71	1299	2014	21	177	6305
2013	30	66	1365	2014	22	186	6491
2013	31	69	1434	2014	23	197	6688
2013	32	66	1500	2014	24	198	6886
2013	33	73	1573	2014	25	222	7108
2013	34	70	1643	2014	26	210	7318
2013	35	80	1723	2014	27	199	7517
2013	36	65	1788	2014	28	223	7740
2013	37	71	1859	2014	29	215	7955
2013	38	84	1943	2014	30	228	8183
2013	39	92	2035	2014	31	234	8417
2013	40	81	2116	2014	32	189	8606
2013	41	88	2204	2014	33	250	8856
2013	42	74	2278	2014	34	259	9115
2013	43	97	2375	2014	35	266	9381
2013	44	92	2467				

- User Growth = Number of active users per week
- Query :
- select count(*) from tutorial.yammer_users

  where state = 'active';
- OUTPUT :-
- there are in total 9381 active users from 1st week of 2013 to the 35th week of 2014

count
9381

- Weekly Retention: Users getting retained weekly after signing-up for a product.Calculate the weekly retention of users-sign up cohort?
- The weekly retention of users-sign up cohort can be calculated by two means i.e. either by specifying the week number (18 to 35) or for the entire column of occurred_at of the events table.
- Firstly we will extract the week from occurred_at column using the extract,week functions
- Then, we will select out those rows in which event_type = 'signup_flow' and event_name = 'complete_signup'
- If finding for a spectifc week we will spectify the week number using the extract function
- Then using the left join we will join the two tables on the basis of user_id where event_type = 'engagement'
- Then we will use the Group By function to group the output table on the basis of
- user_id
- Then we will use the Order By function to order the result table on the basis of user_id

- QUERY :-
- SELECT distinct user_id, COUNT(user_id), SUM(CASE WHEN retention_week = 1 Then 1 Else 0 END) as per_week_retention FROM (SELECT a.user_id, a.signup_week, b.engagement_week, b.engagement_week - a.signup_week as retention_week FROM ((SELECT distinct user_id, extract(week from occurred_at) as signup_week from tutorial.yammer_events WHERE event_type = 'signup_flow' and event_name = 'complete_signup')a LEFT JOIN (SELECT distinct user_id, extract (week from occurred_at) as engagement_week FROM tutorial.yammer_events where event_type = 'engagement')b on a.user_id = b.user_id))d group by user_id order by user_id;
- OUTPUT :-
- LINK :- https://drive.google.com/file/d/1mr8dFvW8et3VBKlnA9e4XmXoGWvUtgTu/view?usp=share_link

- Weekly Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly. Calculate the weekly engagement per device?
- To find the weekly user engagement per device:-
- Firstly we will extract the year_num and week_num from the occurred_at column of the events table using the extract, year and week function
- Then we will select those rows where event_type = 'engagement' using the WHERE clause
- Then by using the Group By and Order By function we will group and order the result on the basis of year_num, week_num and device

- QUERY :-
- SELECT
- extract(year from occurred_at) as year_num,
- extract(week from occurred_at) as week_num,
- device, COUNT(distinct user_id) as no_of_users FROM tutorial.yammer_events where event_type = 'engagement' GROUP by 1,2,3 order by 1,2,3;
- OUTPUT :-
- LINK :- https://drive.google.com/file/d/1kvv1VpqsZBuUyDCYqdVfC2_pL4GvHS_K/view?usp=sharing

- Email Engagement: Users engaging with the email service. Calculate the email engagement metrics?
- To find the email engagement metrics(rate) of users:-
- We will first categorize the action on the basis of email_sent, email_opened and email_clicked using the CASE, WHEN, THEN functions
- Then we select the sum of category of email_opened divide by the sum of the category of email_sent and multiply the result by 100.0 and name is as email_opening_rate
- Then we select the sum of category of email_clicked divide by the sum of the category of email_sent and multiply the result by 100.0 and name is as email_clicking_rate
- email_sent = ('sent_weekly_digest','sent_reengagement_email')
- email_opened = 'email_open'
- email_clicked = 'email_clickthrough'

- QUERY :-
- SELECT
- 100.0*SUM(CASE when email_cat = 'email_opened' then 1 else 0 end)/SUM(CASE when
- email_cat = 'email_sent' then 1 else 0 end) as email_opening_rate,
- 100.0*SUM(CASE when email_cat = 'email_clicked' then 1 else 0 end)/SUM(CASE when
- email_cat = 'email_sent' then 1 else 0 end) as email_clicking_rate FROM (SELECT *, CASE WHEN action in ('sent_weekly_digest','sent_reengagement_email') then 'email_sent' WHEN action in ('email_open') then 'email_opened' WHEN action in ('email_clickthrough') then 'email_clicked' end as email_cat from tutorial.yammer_emails) a;

- OUTPUT :-

	A	B
1	email_opening_rate	email_clicking_rate
2	33.58338805	14.78988638

THE END