Himanshu Jain
Himjain567@gmail.com

Operation Analytics and Investing Metric Spike

Description:

Operation analytics is the subset of data analytics that mainly focuses on the improving efficiency of the operations in the company. It shows how currently different operations are working and how those operations can be improved for better profits. In this project, I have answered such questions which are generally asked by the different departments in a company like the ops team, support team, marketing team, etc. which are required to increase efficiency and streamline.

Investigating metric spikes is an important part of operation analytics. The metric spike shows the anomaly in the trends. It gives the answers to questions like- why there is a dip in daily engagement. why have sales taken a dip? etc. These questions are to be answered daily or weekly basis and should be treated seriously. For this task, different data sets are given.

Approach:

To start with the project first I understood all the problem statements. Tried to find out what tables will be required to find the best possible result and marked it to use while actual query writing. The queries should be easy to understand. I have written each of the primary and foreign keys for the tables. So, at the time of writing the query, I did have not to check again and again to get those columns. Studied unknown terms like throughput.

Tech-Stack used:

To solve these problems, I have used MYSQL Workbench 8.0 CE. Which is an open software and can be downloaded from https://www.mysql.com/.

Insights:

The project is extremely helpful to understand the basics of MySQL. It helped me to learn the structure. It also helped me to learn new keywords like week, day, etc. I have also learned the concept of BETWEEN, GROUP BY, ORDER BY, CASE, Window function, partition by, over, rows between, etc. This project gave me the confidence to work in SQL.

Also, I have learned to import CSV files in the MYSQL workbench. But the files consist of a high number of rows which took a lot of time.

Himanshu Jain
Himjain567@gmail.com

Results:

Case Study 1 (Job Data):

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

```
/* Calculate the number of jobs reviewed per hour per day
for november 2020? */
use ig_clone;
select ds,
1.0*count(job_id)*3600/sum(time_spent) as jobs_reviewed_per_day_per_hour
from job_data
where event in ('transfer','decision') and extract(month from ds)=11
group by ds
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	jobs_reviewed_per_day_per_hour
2020-11-30 00:00:00	144.00000
2020-11-29 00:00:00	180.00000
2020-11-28 00:00:00	218.18182
2020-11-27 00:00:00	34.61538
2020-11-25 00:00:00	80.00000

B. Throughput: It is the no. of events happening per second. Let's say the above metric is called throughput. For throughput, do you prefer daily metric or 7-day rolling, and why?

If the density of the data is larger, we use the daily metric, and if the density is low the 7-day rolling works well. It also depends upon the anomaly in the data set. Because in 7 days metric the view is broader similar to the daily metric view becomes narrower.

Himanshu Jain
Himjain567@gmail.com

```sql
/* Calculate 7 day rolling average of throughtput? */
select ds, round(1*sum(count(job_id))over(order by ds rows between 6
preceding and current row) /(sum(sum(time_spent))
over (order by ds rows between 6 preceding and current row)),2)
as throughtput_7d
from job_data
where event in ('transfer','decision')
group by ds
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	throughtput_7d
2020-11-25 00:00:00	0.02
2020-11-27 00:00:00	0.01
2020-11-28 00:00:00	0.02
2020-11-29 00:00:00	0.02
2020-11-30 00:00:00	0.03

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

```sql
/* Calculate the percentage share of each language in the lsat 30 days? */
select language, count(job_id) as job_count, count(job_id)*100/total_jobs
as per_share from job_data
cross join (select count(job_id) as total_jobs from job_data) a
group by language
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

language	job_count	per_share
English	1	12.5000
Arabic	1	12.5000
Persian	3	37.5000
Hindi	1	12.5000
French	1	12.5000
Italian	1	12.5000

Himanshu Jain
Himjain567@gmail.com

D. Duplicate rows: Rows that have the same value present in them.
When the row has been duplicated the value of the 'rownum' column in the output will be greater than 1. And the duplicate rows can be extracted by removing the comment in the where clause.

```
/* How will you display duplicate rows? */
select ds,job_id,event,language,time_spent,org,rownum
from (select *,row_number() over(partition by ds, job_id,actor_id) as rownum
from job_data) a
/* where rownum>1 */
```

Result Grid

ds	job_id	event	language	time_spent	org	rownum
2020-11-25 00:00:00	20	transfer	Italian	45	C	1
2020-11-26 00:00:00	23	skip	Persian	56	A	1
2020-11-27 00:00:00	11	decision	French	104	D	1
2020-11-28 00:00:00	23	transfer	Persian	22	D	1
2020-11-28 00:00:00	25	decision	Hindi	11	B	1
2020-11-29 00:00:00	23	decision	Persian	20	C	1
2020-11-30 00:00:00	21	skip	English	15	A	1
2020-11-30 00:00:00	22	transfer	Arabic	25	B	1

Himanshu Jain
Himjain567@gmail.com

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.

```
/* Calculate the weekly user engagement */
SELECT week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')) as week_num,
count(distinct user_id) as user_engagement from events
where event_type ='engagement' and
event_name='login'
group by 1
order by 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

week_num	user_engagement
17	663
18	1068
19	1113
20	1154
21	1121
22	1186
23	1232
24	1275
25	1264
26	1302
27	1372
28	1365
29	1376
30	1467
31	1299
32	1225
33	1225
34	1204
35	104

Himanshu Jain
Himjain567@gmail.com

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

```
/* Calculate user growth for product.*/
select day(created_at) as day,
count(*) as all_users,
count(activated_at) as activated_users
from users u
where created_at>='2013-03-01'
and created_at<'2013-03-31'
group by 1 order by 1
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

day	all_users	activated_users
1	15	8
2	4	1
3	4	0
4	18	9
5	13	7
6	15	7
7	15	9
8	15	8
9	4	3
10	5	0
11	16	8
12	17	5
13	15	6
14	16	8
15	15	4
16	4	1
17	4	1
18	17	6
19	17	4
20	17	5
21	18	7
22	18	8
23	4	0
24	4	2
25	19	7
26	17	7

Himanshu Jain
Himjain567@gmail.com

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

```sql
/* Calculate the weekly retention of user=-sign up cohort. */
select first_week,
sum(case when week_num=1 then 1 else 0 end) as week_0,
sum(case when week_num=2 then 1 else 0 end) as week_1,
sum(case when week_num=3 then 1 else 0 end) as week_2,
sum(case when week_num=4 then 1 else 0 end) as week_3,
sum(case when week_num=5 then 1 else 0 end) as week_4,
sum(case when week_num=6 then 1 else 0 end) as week_5,
sum(case when week_num=7 then 1 else 0 end) as week_6,
sum(case when week_num=8 then 1 else 0 end) as week_7,
sum(case when week_num=9 then 1 else 0 end) as week_8,
sum(case when week_num=10 then 1 else 0 end) as week_9,
sum(case when week_num=11 then 1 else 0 end) as week_10
from (select a.user_id, week, first_week,(week-first_week)
as week_num from
(select user_id, week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')) as week
from events group by user_id, week) a,
(select user_id, min(week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')))
as first_week
from events group by user_id) b
where a.user_id=b.user_id) as with_week_number
group by first_week order by first_week;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_week	week_0	week_1	week_2	week_3	week_4	week_5	week_6	week_7	week_8	week_9	week_10
17	472	324	251	205	187	167	146	145	145	136	131
18	362	261	203	168	147	144	127	113	122	106	118
19	284	173	153	114	95	91	81	95	82	68	65
20	223	165	121	91	72	63	67	63	65	67	41
21	187	131	91	74	63	75	72	58	48	45	39
22	224	150	107	87	73	63	60	55	48	41	39
23	219	138	101	90	79	69	61	54	47	35	30
24	205	143	102	81	63	65	61	38	39	29	0
25	218	139	101	75	63	50	46	38	35	2	0
26	181	114	83	73	55	47	43	29	0	0	0
27	199	121	106	68	53	40	36	1	0	0	0
28	194	114	69	46	30	28	3	0	0	0	0
29	186	102	65	47	40	1	0	0	0	0	0
30	202	121	78	53	3	0	0	0	0	0	0
31	145	76	57	1	0	0	0	0	0	0	0
32	188	94	8	0	0	0	0	0	0	0	0
33	202	9	0	0	0	0	0	0	0	0	0

Himanshu Jain
Himjain567@gmail.com

D. 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 */
select first_week,weekly_number.device,
sum(case when week_num=1 then 1 else 0 end) as week_0,
sum(case when week_num=2 then 1 else 0 end) as week_1,
sum(case when week_num=3 then 1 else 0 end) as week_2,
sum(case when week_num=4 then 1 else 0 end) as week_3,
sum(case when week_num=5 then 1 else 0 end) as week_4,
sum(case when week_num=6 then 1 else 0 end) as week_5,
sum(case when week_num=7 then 1 else 0 end) as week_6,
sum(case when week_num=8 then 1 else 0 end) as week_7,
sum(case when week_num=9 then 1 else 0 end) as week_8,
sum(case when week_num=10 then 1 else 0 end) as week_9,
sum(case when week_num=11 then 1 else 0 end) as week_10
from (select a.user_id,device, week, first_week,(week-first_week)
as week_num from
(select user_id,device, week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i'))
as week
from events where event_type='engagement' group by user_id, week,device) a,
(select user_id, min(week(STR_TO_DATE(occurred_at, '%m/%d/%Y %H:%i')))
as first_week from events group by user_id) b
where a.user_id=b.user_id) as weekly_number
group by 1,2 order by 1,2
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_week	device	week_0	week_1	week_2	week_3	week_4	week_5	week_6	week_7	week_8
27	kindle fire	5	3	1	0	0	0	0	0	0
27	lenovo thinkpad	28	20	11	7	9	7	8	0	0
27	mac mini	4	2	2	2	1	0	0	0	0
27	macbook air	31	12	10	8	8	2	6	0	0
27	macbook pro	52	32	20	17	11	9	10	0	0
27	nexus 10	2	2	3	1	2	1	3	0	0
27	nexus 5	13	8	5	4	1	1	1	0	0
27	nexus 7	7	1	5	3	1	2	0	0	0
27	nokia lumia 635	2	2	0	0	0	0	1	0	0
27	samsumg galaxy tablet	3	2	1	1	1	0	1	0	0
27	samsung galaxy note	0	0	0	0	0	0	0	0	0
27	samsung galaxy s4	18	9	11	2	4	2	1	0	0
27	windows surface	3	4	0	3	0	0	0	0	0
28	acer aspire desktop	4	1	1	1	2	1	0	0	0
28	acer aspire notebook	8	4	4	3	0	1	0	0	0
28	amazon fire phone	2	1	0	0	0	0	0	0	0
28	asus chromebook	8	5	3	4	1	1	0	0	0

Himanshu Jain
Himjain567@gmail.com

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

```sql
/* Calculate the email engagement metrics */
select week(occurred_at) as week,
count(case when action='sent_weekly_digest' then user_id else null end)
as weekly_digest,
count(case when action='email_open' then user_id else null end)
as email_open,
count(case when action='email_clickthrough' then user_id else null end)
as email_clickthrough,
count(case when action='sent_reengagement_email' then user_id else null end)
as reengagement_email
from email_events
group by 1
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

week	weekly_digest	email_open	email_clickthrough	reengagement_email
18	2602	912	430	157
19	2665	972	477	173
20	2733	1004	507	191
21	2822	1014	443	164
22	2911	987	488	192
23	3003	1075	538	197
24	3105	1155	554	226
25	3207	1096	530	196
26	3302	1165	556	219
27	3399	1228	621	213
28	3499	1250	599	213
29	3592	1219	590	213
30	3706	1383	630	231
31	3793	1351	445	222
32	3897	1337	418	200
33	4012	1432	490	264
34	4111	1528	490	261
17	908	310	166	73
35	0	41	38	48