Operation Analytics and Investigating Metric Spike

(Advanced SQL)

Project Description

Operation Analytics is the analysis done for the complete end to end operations of a company. With the help of this, the company then finds the areas on which it must improve upon. You work closely with the ops team, support team, marketing team, etc and help them derive insights out of the data they collect.

Being one of the most important parts of a company, this kind of analysis is further used to predict the overall growth or decline of a company's fortune. It means better automation, better understanding between cross-functional teams, and more effective workflows.

Investigating metric spike is also an important part of operation analytics as being a Data Analyst you must be able to understand or make other teams understand questions like- Why is there a dip in daily engagement? Why have sales taken a dip? Etc. Questions like these must be answered daily and for that its very important to investigate metric spike.

We have been provided two datasets , Case Study-1 Dataset(Job_Data) and Case Study-2 Dataset (Investigating Metric Spike) and required to provide a detailed report for the below two operations mentioning the answers for the related questions:

Case Study 1 (Job Data)

Below is the structure of the table with the definition of each column that you must work on:

- **Table-1:** job_data
    - **job_id:** unique identifier of jobs
    - **actor_id:** unique identifier of actor
    - **event:** decision/skip/transfer
    - **language:** language of the content
    - **time_spent:** time spent to review the job in seconds
    - **org:** organization of the actor
    - **ds:** date in the yyyy/mm/dd format. It is stored in the form of text and we use presto to run. no need for date function

Use this dataset answer the questions that follows

1. **Number of jobs reviewed:** Amount of jobs reviewed over time.
   My task: Calculate the number of jobs reviewed per hour per day for November 2020?
2. **Throughput:** It is the no. of events happening per second.
   My 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?
3. **Percentage share of each language:** Share of each language for different contents.
   My task: Calculate the percentage share of each language in the last 30 days?
4. **Duplicate rows:** Rows that have the same value present in them.
   My 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)

- **Table-1:** users
  This table includes one row per user, with descriptive information about that user's account.
- **Table-2:** events
  This table includes one row per event, where an event is an action that a user has taken. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.
- **Table-3:** email_events
  This table contains events specific to the sending of emails. It is similar in structure to the events table above.

Using this dataset answer the questions that follows

1. **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
   My task: Calculate the weekly user engagement?
2. **User Growth:** Amount of users growing over time for a product.
   My task: Calculate the user growth for product?
3. **Weekly Retention:** Users getting retained weekly after signing-up for a product.
   My task: Calculate the weekly retention of users-sign up cohort?
4. **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
   My task: Calculate the weekly engagement per device?
5. **Email Engagement:** Users engaging with the email service.
   My task: Calculate the email engagement metrics?

Approach

First of all I have imported the dastabase on my MySQL Workbench. Then I analyzed the database carefully. Observing all the tables, columns, rows, and relationship among all the table, and created ER Diagram of complete database provided.

Before finding the answers of the questions I need to have the data understanding of the database provided as well as the business understanding. Then I have done Data Profiling and created a Data Model like numbers of rows and columns we have in every Table, Datatypes,Keys,Relationships.

After doing all this , I started to find answers of the questions provided to me by the Operations Team by Querying the database.

Tech-Stack Used

I have used MySQL Workbench v8.0.31 by Oracle for project execution in order to query the database. The ease of access and setup, troubleshooting support as well as the GUI made it a good tool for the project.

Insights

Case Study 1 (Job Data)

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

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

QUERY:-

```
SELECT COUNT(DISTINCT job_id)/(30*24) as jobrev_prhr_prday
FROM job_data where ds like '2020-11%';
```

OUTPUT:-

	jobrev_prhr_prday
▶	0.0083

CONCLUSION:- **Here , Number of jobs reviewed per hour per day in the month of November 2020 is 0.0083.**

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

My 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?

QUERY:-

```
DROP TABLE IF EXISTS JOBS_REVIEWED;
CREATE TEMPORARY TABLE JOBS_REVIEWED
(
SELECT ds,count(distinct job_id) as jobs_reviewed
,CAST(COUNT(DISTINCT JOB_ID)/86400 AS DECIMAL(10,10)) AS THROUGHPUT
    from job_data
    group by ds
    order by ds
);
SELECT ds,jobs_reviewed,throughput
,avg(throughput) OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS throughput_7day
FROM JOBS_REVIEWED;
```

OUTPUT:-

	ds	jobs_reviewed	throughput	throughput_7day
▶	2020-11-25	1	0.0000115740	0.00001157400000
	2020-11-26	1	0.0000115740	0.00001157400000
	2020-11-27	1	0.0000115740	0.00001157400000
	2020-11-28	2	0.0000231480	0.00001446750000
	2020-11-29	1	0.0000115740	0.00001388880000
	2020-11-30	2	0.0000231480	0.00001543200000

CONCLUSION:-

I think 7-Day Rolling Average would be much better than daily metric for better performance of any business , as it relates with only the recent trends and help us to compare which group of days perform better and it helps to understand why it is so, thus we can stay in trend and keep updating ourselves.

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

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

QUERY:-

```
SELECT job_id,language,count(language) AS LANG_COUNT,
ROUND((COUNT(LANGUAGE)/6),2)*100 AS PERCENTAGE
FROM job_data
GROUP BY job_id,language;
```

OUTPUT:-

	job_id	language	LANG_COUNT	PERCENTAGE
▶	21	English	1	17.00
	22	Arabic	1	17.00
	23	Persian	3	50.00
	25	Hindi	1	17.00
	11	French	1	17.00
	20	Italian	1	17.00

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

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

QUERY:-

```
SELECT ds,a.job_id,language
FROM job_data
```

```
INNER JOIN
(SELECT job_id,count(job_id) as dup_id
FROM job_data
GROUP BY job_id
HAVING dup_id>1
) a
ON a.job_id=job_data.job_id;
```

OUTPUT:-

	ds	job_id	language
▶	2020-11-29	23	Persian
	2020-11-28	23	Persian
	2020-11-26	23	Persian

CONCLUSION:- here we can say that job_id 23 is a duplicate value which comes 3 times in different dates.

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.

My task: Calculate the weekly user engagement?

QUERY:-

```
SELECT WEEK(str_to_date(occured_at,'%Y-%m-%d')) as weekNumber,
        count(user_id) as UserCount
        from events
        group by weekNumber;
```

OUTPUT:-

	weekNumber	UserCount
▶	17	8404
	18	18242
	19	18178
	20	18866
	21	18112
	23	19345
	22	19455
	24	20210
	25	19717
	29	21233
	26	20126
	30	22775
	28	21908
	27	21021
	31	19585
	32	17872
	33	17445
	34	17466
	35	872

CONCLUSION:-

We can see here the Week Number and the number of users active in that week. In this way we can measure the weekly engagement of users.

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

My task: Calculate the user growth for product?

QUERY:-

```
SELECT WEEK(STR_TO_DATE(created_at,'%Y-%m-%d')) AS week_num,
COUNT(user_id) NoOfUsers,
COUNT(USER_ID) - LAG(COUNT(user_id),1) OVER(ORDER BY WEEK(STR_TO_DATE(created_at,'%Y-%m-%d')))
  AS user_growth
FROM users
GROUP BY week_num
order by week_num;
```

OUTPUT:-

week_num	NoOfUsers	user_growth
0	197	NULL
1	300	103
2	299	-1
3	325	26
4	322	-3
5	341	19
6	344	3
7	353	9
8	350	-3
9	353	3
10	377	24
11	382	5
12	391	9
13	396	5
14	411	15
15	395	-16
16	465	70
17	450	-15
18	460	10
19	467	7
20	477	10
21	474	-3
22	503	29
23	526	23
24	543	17

25	529	-14
26	535	6
27	555	20
28	568	13
29	582	14
30	618	36
31	539	-79
32	622	83
33	625	3
34	647	22
35	196	-451
36	164	-32
37	164	0
38	166	2
39	180	14
40	174	-6
41	172	-2
42	191	19
43	195	4
44	194	-1
45	191	-3
46	179	-12
47	207	28
48	213	6
49	216	3
50	221	5
51	235	14
52	87	-148

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

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

QUERY:-

```
SELECT
COUNT(user_id),
SUM(CASE
WHEN retention_week = 1 THEN 1
ELSE 0
END) AS week_1
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 occured_at) AS signup_week
FROM
events
WHERE
event_type = 'signup_flow'
AND event_name = 'complete_signup'
AND EXTRACT(WEEK FROM occured_at) = 18) a
LEFT JOIN (SELECT DISTINCT
user_id, EXTRACT(WEEK FROM occured_at) AS engagement_week
FROM
events
WHERE event_type = 'engagement') b
ON a.user_id = b.user_id)
ORDER BY a.user_id) a;
```

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

My task: Calculate the weekly engagement per device?

QUERY:-

```
SELECT device, WEEK(STR_TO_DATE(occured_at,'%Y-%m-%d'))AS week_num,

COUNT(user_id) as total_users

FROM events

WHERE event_type='engagement'

GROUP BY device,week_num

ORDER BY week_num DESC;
```

OUTPUT:-

device	week_num	total_users
acer aspire desktop	35	7
acer aspire notebook	35	28
asus chromebook	35	38
dell inspiron desktop	35	4
dell inspiron notebook	35	66
hp pavilion desktop	35	10
htc one	35	18
ipad mini	35	21
iphone 4s	35	57
iphone 5	35	9
iphone 5s	35	22

491 Rows Returned.

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

My task: Calculate the email engagement metrics?

QUERY:-

```
SELECT
100.0 * SUM(CASE
WHEN email_cat = 'email_open' THEN 1
ELSE 0
END) / SUM(CASE
WHEN email_cat = 'email_sent' THEN 1
ELSE 0
END) AS email_open_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_clicked_rate
FROM
(SELECT
*,
CASE
WHEN action IN ('sent_weekly_digest' , 'sent_reengagement_email') THEN
'email_sent'
WHEN action IN ('email_open') THEN 'email_open'
WHEN action IN ('email_clickthrough') THEN 'email_clicked'
END AS email_cat
FROM
email_events) a;
```

OUTPUT:-

	email_open_rate	email_clicked_rate
▶	33.58339	14.78989

Result

Working on this project I have got the real experience of working on corporate live projects. Thanks to the Trainity for giving me this opportunity.

I also able to use my concepts like Window Functions, Joins, Common Table Expression , Group By, Aggregate functions , Operators etc. which I learned and explore those concepts in different ways.

Prepared by: Raushan Kumar Chaurasiya

POSITION	Data Analyst Trainee
DD/MM/YYYY	28/01/2023