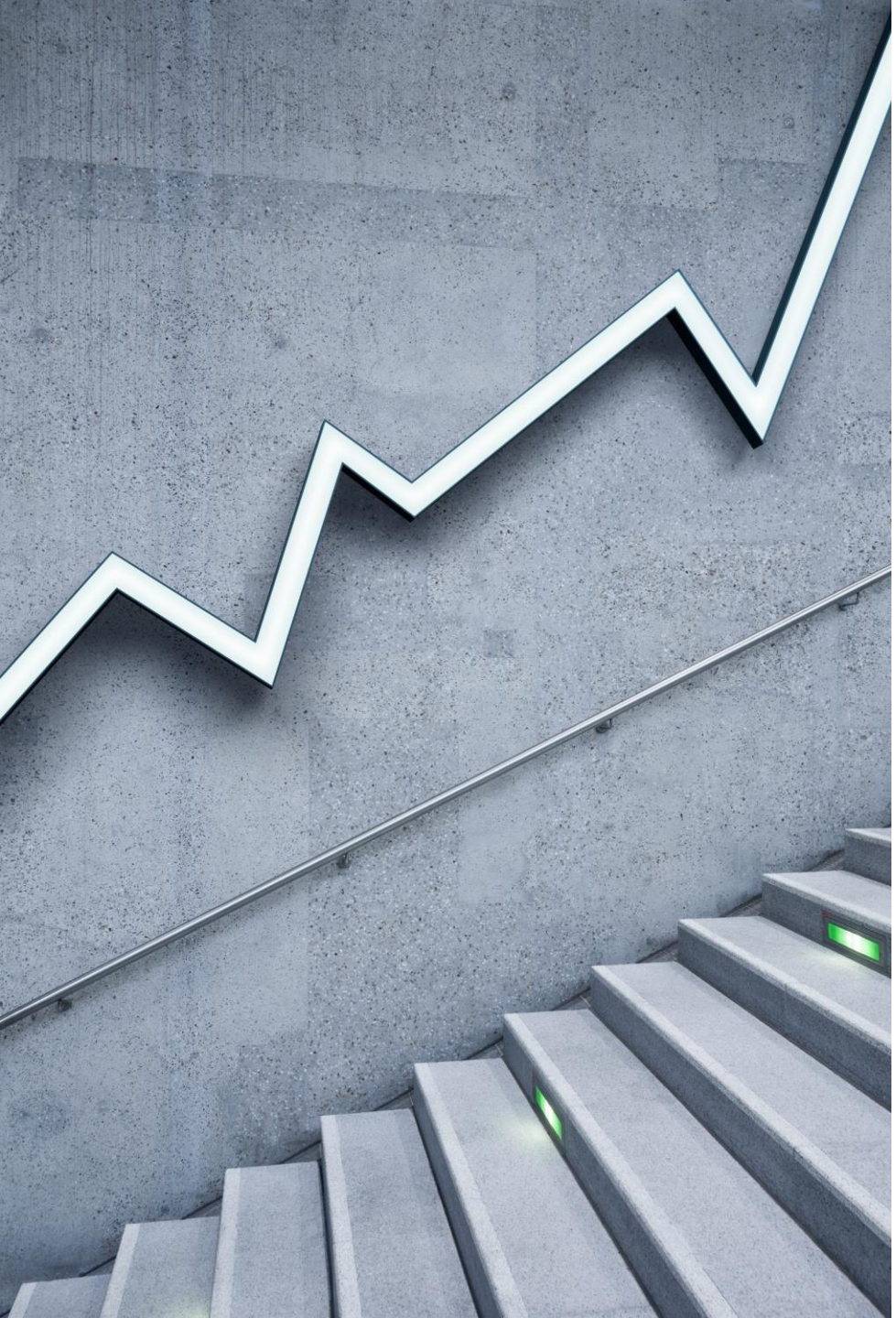

Operation Analytics and Investigating Metric Spike

BY: RAJAT PANWAN

Project Description

The project is about analyzing two different data sets related to the end-to-end operations of a company. As a Data Analyst Lead at Microsoft, the aim is to derive insights from the given data sets and answer the questions asked by different departments of the company. The first case study is related to job data and requires the calculation of various metrics, such as the number of jobs reviewed, throughput, percentage share of each language, and displaying duplicate rows. The second case study is related to investigating metric spikes and requires the calculation of various metrics, such as weekly user engagement, user growth, weekly retention, weekly engagement per device, and email engagement metrics.

Approach

To complete the project, I go through the data provided in each table and understood the relationships between them. Next, I created a database "metric_spike" and tables using SQL queries. Once the tables are created, I started performing the analysis by writing SQL queries to answer the questions asked in each case study. Finally, compiled the results and insights gained from the analysis.

Tech Stack Used

A database management system MySQL 8.0 is used to handle, store and modify and delete data and also store data in an organized way In this process MySQL Workbench in used which comes with MySQL

Insights

Here are some insights and knowledge that I gained while working on Instagram User Analytics project such as understating of the SQL language and how to use it to retrieve and manipulate data in a database Develop an ability to design and execute complex queries using a range of SQL clauses, functions and operators Skills in data analysis and problem solving as the process of creating an SQL query often involves identifying patterns in the data.

Result:

Result of the queries I write is on the next pages

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?

SQL Query:

```sql
SELECT
    COUNT(*) / (24*30) AS job
FROM
    job_data
WHERE
    ds >= '2020-11-01' AND ds < '2020-12-01';
```

Result:

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

job
0.0111

Case Study 1 (Job Data):

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?

SQL Query:

```sql
SELECT ds, AVG(event_per_second)
OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS rolling_avg
FROM (
SELECT ds, COUNT(event) / SUM(time_spent) AS event_per_second
FROM job_data
GROUP BY ds
)rolling_avg_tbl;
```

Result :

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

ds	rolling_avg
2020-11-25	0.02220000
2020-11-26	0.02005000
2020-11-27	0.01656667
2020-11-28	0.02757500
2020-11-29	0.03206000
2020-11-30	0.03505000

Case Study 1 (Job Data):

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?

SQL Query:

```sql
SELECT language, COUNT(job_id), 100 * COUNT(job_id) / SUM(COUNT(job_id))
OVER() AS language_share
FROM job_data
GROUP BY language;
```

Result :

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

language	COUNT(job_id)	language_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

Case Study 1 (Job Data):

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?

SQL Query:

```sql
SELECT *
FROM job_data
WHERE (ds, job_id, actor_id, event, language, time_spent, org) IN (
    SELECT ds, job_id, actor_id, event, language, time_spent, org
    FROM job_data
    GROUP BY ds, job_id, actor_id, event, language, time_spent, org
    HAVING COUNT(*) > 1
    )
ORDER BY ds DESC, job_id DESC;
```

Result :

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

ds	job_id	actor_id	event	language	time_spent	org

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?

SQL Query:

```sql
SELECT
    user_id,
    ((COUNT(*)) / (COUNT(DISTINCT (EXTRACT(WEEK FROM occurred_at))))) AS user_engagement_per_week
FROM
    events
GROUP BY user_id
ORDER BY user_id;
```

Result :

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

user_id	user_engagement_per_week
4	10.3333
8	7.2000
11	31.5000
17	27.5000
19	14.2000
20	11.3333
22	44.6250
30	8.3750
49	5.0000
59	42.8333

Result 30

Case Study 2 (Investigating metric spike):

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

Your task: Calculate the user growth for product?

SQL Query:

```sql
SELECT
  YEAR(created_at) AS year,
  MONTH(created_at) AS month,
  COUNT(DISTINCT user_id) AS user_counts,
  (COUNT(DISTINCT user_id)
    / LAG(COUNT(DISTINCT user_id)) OVER (ORDER BY YEAR(created_at), MONTH(created_at)) - 1)*100
    AS user_percentage_growth
FROM users
GROUP BY year, month;
```

Result :

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

year	month	user_counts	user_percentage_growth
2013	1	332	NULL
2013	2	328	-1.2048
2013	3	383	16.7683
2013	4	410	7.0496
2013	5	486	18.5366
2013	6	485	-0.2058
2013	7	608	25.3608
2013	8	636	4.6053
2013	9	699	9.9057
2013	10	826	18.1688

Result 9

Case Study 2 (Investigating metric spike):

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

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

SQL Query:

```sql
SELECT EXTRACT(WEEK FROM occurred_at) AS weeks,
       COUNT(CASE WHEN e.event_type = 'signup_flow' THEN e.user_id ELSE NULL END) AS signup
FROM events e
GROUP BY weeks
ORDER BY weeks;
```

Result :

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

weeks	signup
17	385
18	901
19	954
20	955
21	961
22	1042
23	1065
24	1158
25	1075
26	1065

Result 8

Case Study 2 (Investigating metric spike):

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?

SQL Query:

```sql
SELECT
    device,
    (COUNT(event_name) /
    (COUNT(DISTINCT(EXTRACT(WEEK FROM occurred_at))))) AS weekly_avg_engagement_per_device
FROM
    events
GROUP BY device
ORDER BY weekly_avg_engagement_per_device DESC;
```

Result :

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

device	weekly_avg_engagement_per_device
macbook pro	3155.1579
lenovo thinkpad	2035.7368
macbook air	1479.1579
iphone 5	1428.1053
dell inspiron notebook	1077.6842
samsung galaxy s4	1031.2632
nexus 5	907.8421
iphone 5s	879.2105
dell inspiron desktop	556.2632
iphone 4s	531.4211

Result 13

Case Study 2 (Investigating metric spike):

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

Your task: Calculate the email engagement metrics?

SQL Query:

```sql
SELECT
  user_id,
  COUNT(*) AS email_events_count,
  SUM(CASE WHEN action = 'email_open' THEN 1 ELSE 0 END) AS email_opens_count,
  SUM(CASE WHEN action = 'email_clickthrough' THEN 1 ELSE 0 END) AS email_clickthrough_count,
  SUM(CASE WHEN action = 'sent_weekly_digest' THEN 1 ELSE 0 END) AS sent_weekly_digest_count,
  SUM(CASE WHEN action = 'sent_reengagement_email' THEN 1 ELSE 0 END) AS sent_reengagement_email_count
FROM email_events
GROUP BY user_id;
```

Result :

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

user_id	email_events_count	email_opens_count	email_clickthrough_count	sent_weekly_digest_count	sent_reengagement_email_count
0	22	5	0	17	0
4	26	5	4	17	0
8	21	3	1	17	0
11	24	5	2	17	0
17	22	4	1	17	0
19	23	5	1	17	0
20	28	8	3	17	0
22	27	7	3	17	0
30	25	6	1	18	0
49	23	5	1	17	0

Result 3

Thank You

You can connect me on

rajatpawan@gmail.com