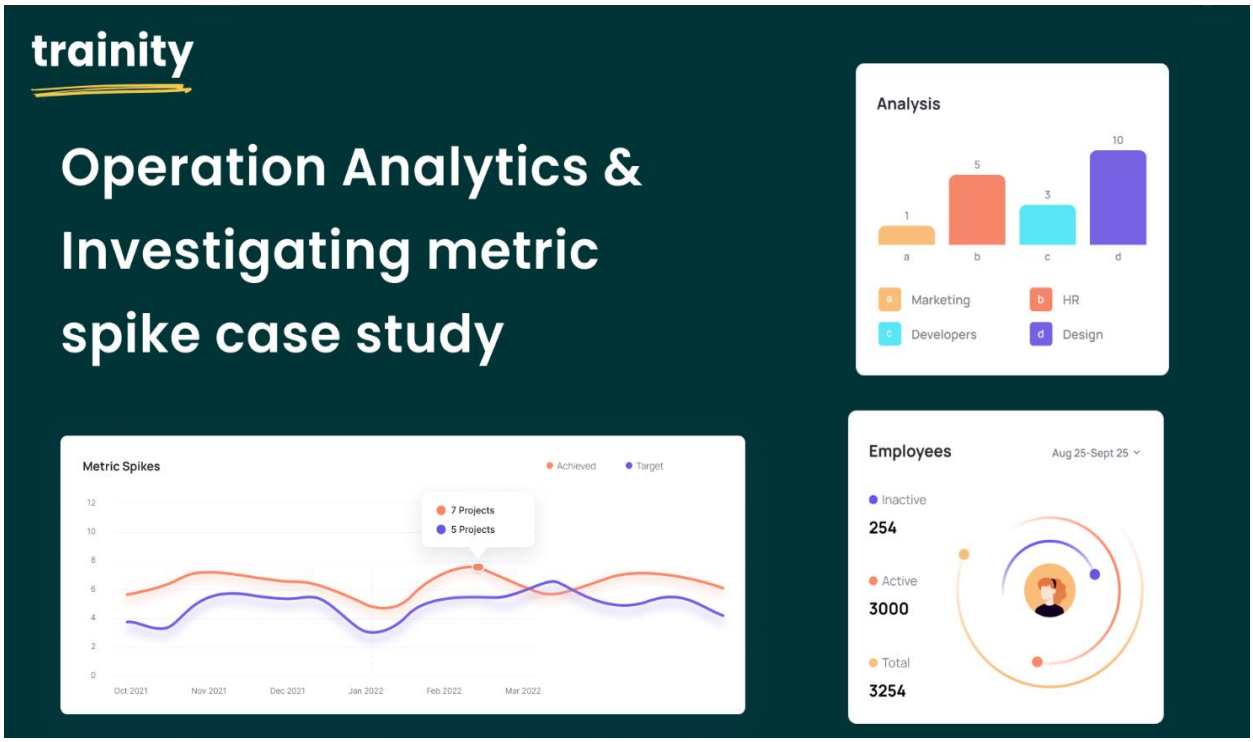
Operation Analytics and Investigating Metric Spike

by Vikas Bhandari

[LinkedIn]

[Github]



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.

I'm working for a company like Microsoft designated as Data Analyst Lead and is provided with different data sets, tables from which you must derive certain insights out of it and answer the questions asked by different departments.

From this project, I have to find answers to some business questions which are below mentioned:

1. **Case Study 1 (Job data)**
    - **Calculate the number of jobs reviewed per hour per day for November 2020?**
    - **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?**
    - **Calculate the percentage share of each language in the last 30 days?**
    - **Calculate the percentage share of each language in the last 30 days?**

2. **Case Study 2 (Investigating Metric Spike)**
    - **Calculate the weekly user engagement?**
    - **Calculate the user growth for product?**
    - **Calculate the weekly retention of users-sign up cohort?**
    - **Calculate the weekly engagement per device?**
    - **Calculate the email engagement metrics?**

Approach

Initially, I dedicated a considerable amount of time to comprehend the data/table that was provided. Subsequently, I proceeded to create a database for the 1st case study based on the provided table. And for the 2nd case study, I utilized the https://tableconvert.com/ website, which facilitated the conversion of the CSV table data into SQL queries. Moving forward, I created the database for the 2nd case study. Finally, I conducted an analysis to uncover the desired answers and insights as requested by the company.

CASE STUDY 1: JOB DATA

1. **Calculate the number of jobs reviewed per hour per day for November 2020?**

SELECT ds AS Date, 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;

Dates	Jobs Resets all sorted c 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

2. **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?**

```
SELECT a.ds AS day,
a.throughput,
round(avg(a.throughput) OVER ( ORDER BY ds rows BETWEEN 6 PRECEDING AND
CURRENT row), 2) AS
7_day_avg_throughput
FROM
( SELECT ds, count(job_id) / sum(time_spent) AS throughput
FROM job_data
GROUP BY ds ) AS a
GROUP BY ds;
```

day	throughput	7_day_avg_throughput
2020-11-25	0.0222	0.02
2020-11-26	0.0179	0.02
2020-11-27	0.0096	0.02
2020-11-28	0.0606	0.03
2020-11-29	0.0500	0.03
2020-11-30	0.0500	0.04

3. **Calculate the percentage share of each language in the last 30 days?**

```
SELECT language,
        count(job_id) as no_of_jobs,
        count(job_id)*100 / sum(count(*)) OVER() as percentage_share
FROM  job_data
WHERE ds between '2020-11-01' and '2020-11-30'
GROUP by language;
```

language	no_of_jobs	percentage_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

4. **Calculate the percentage share of each language in the last 30 days?**

```
SELECT
        actor_id,
        COUNT(*) AS Duplicates
FROM job_data
GROUP BY actor_id
HAVING COUNT(*) > 1;
```

actor_id	Duplicates
1004	2

CASE STUDY 2: INVESTIGATING METRIC SPIKE

1. **Calculate the weekly user engagement?**

```
SELECT  WEEK(occurred_at) AS Week,
   COUNT(DISTINCT user_id) AS Weekly_User_engagement
FROM    events
GROUP BY WEEK(occurred_at)
ORDER BY WEEK(occurred_at);
```

Week	Weekly_User_engagement
17	740
18	1260
19	1287
20	1351
21	1299
22	1381
23	22
24	1471
25	1459
26	1509
27	1573
28	1577
29	1607
30	1706
31	1514
32	1454
33	1438
34	1443
35	118

2. **Calculate the user growth for product?**

```
SET @g := 0;
SELECT
   o.no_of_users,
   o.date,
   (@g:=@g + o.no_of_users) AS user_growth
FROM
   (SELECT
      COUNT(user_id) AS no_of_users, DATE(created_at) AS date
   FROM  users
   WHERE
      state = 'active'
   GROUP BY DATE(created_at)) o;
```

no_of_users	date	user_growth
7	2013-01-01	7
7	2013-01-02	14
6	2013-01-03	20
1	2013-01-04	21
2	2013-01-05	23
3	2013-01-06	26
4	2013-01-07	30
2	2013-01-08	32
6	2013-01-09	38
6	2013-01-10	44
6	2013-01-11	50
3	2013-01-12	53
2	2013-01-13	55
8	2013-01-14	63
11	2013-01-15	74
7	2013-01-16	81
9	2013-01-17	90

3. **Calculate the weekly retention of users-sign up cohort?**

```
SELECT first AS "Week Numbers",
SUM(CASE WHEN week_number = 0 THEN 1 ELSE 0 END) AS "Week 0",
SUM(CASE WHEN week_number = 1 THEN 1 ELSE 0 END) AS "Week 1",
SUM(CASE WHEN week_number = 2 THEN 1 ELSE 0 END) AS "Week 2",
SUM(CASE WHEN week_number = 3 THEN 1 ELSE 0 END) AS "Week 3",
SUM(CASE WHEN week_number = 4 THEN 1 ELSE 0 END) AS "Week 4",
SUM(CASE WHEN week_number = 5 THEN 1 ELSE 0 END) AS "Week 5",
SUM(CASE WHEN week_number = 6 THEN 1 ELSE 0 END) AS "Week 6",
SUM(CASE WHEN week_number = 7 THEN 1 ELSE 0 END) AS "Week 7",
SUM(CASE WHEN week_number = 8 THEN 1 ELSE 0 END) AS "Week 8",
SUM(CASE WHEN week_number = 9 THEN 1 ELSE 0 END) AS "Week 9",
SUM(CASE WHEN week_number = 10 THEN 1 ELSE 0 END) AS "Week 10",
SUM(CASE WHEN week_number = 11 THEN 1 ELSE 0 END) AS "Week 11",
SUM(CASE WHEN week_number = 12 THEN 1 ELSE 0 END) AS "Week 12",
SUM(CASE WHEN week_number = 13 THEN 1 ELSE 0 END) AS "Week 13",
SUM(CASE WHEN week_number = 14 THEN 1 ELSE 0 END) AS "Week 14",
```

```
SUM(CASE WHEN week_number = 15 THEN 1 ELSE 0 END) AS "Week 15",
SuM(CASE WHEN week_number = 16 THEN 1 ELSE 0 END) AS "Week 16",
SUM(CASE WHEN week_number = 17 THEN 1 ELSE 0 END) AS "Week 17",
SUM(CASE WHEN week_number = 18 THEN 1 ELSE 0 END) AS "Week 18"
FROM
(
SELECT m.user_id, m.login_week, n.first, m.login_week - first AS week_number
FROM
(SELECT user_id, EXTRACT(WEEK FROM occurred_at) AS login_week FROM events
GROUP BY 1,2) m,
(SELECT user_id, MIN(EXTRACT(WEEK FROM occurred_at)) AS first FROM events
GROUP BY 1) n
WHERE m.user_id = n.user_id
) sub
GROUP BY first
ORDER BY first;
```

Week Numbers	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18
17	740	472	324	251	205	187	167	146	145	145	136	131	132	143	116	91	82	77	5
18	788	362	261	203	168	147	144	127	113	122	106	118	127	110	97	85	67	4	0
19	601	284	173	153	114	95	91	81	95	82	68	65	63	42	51	49	2	0	0
20	555	223	165	121	91	72	63	67	63	65	67	41	40	33	40	0	0	0	0
21	495	187	131	91	74	63	75	72	58	48	45	39	35	28	2	0	0	0	0
22	521	224	150	107	87	73	63	60	55	48	41	39	31	1	0	0	0	0	0
23	542	219	138	101	90	79	69	61	54	47	35	30	0	0	0	0	0	0	0
24	535	205	143	102	81	63	65	61	38	39	29	0	0	0	0	0	0	0	0
25	500	218	139	101	75	63	50	46	38	35	2	0	0	0	0	0	0	0	0
26	495	181	114	83	73	55	47	43	29	0	0	0	0	0	0	0	0	0	0
27	493	199	121	106	68	53	40	36	1	0	0	0	0	0	0	0	0	0	0
28	486	194	114	69	46	30	28	3	0	0	0	0	0	0	0	0	0	0	0
29	501	186	102	65	47	40	1	0	0	0	0	0	0	0	0	0	0	0	0
30	533	202	121	78	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31	430	145	76	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	496	188	94	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	499	202	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	518	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4. **Calculate the weekly engagement per device?**

```
SELECT
   WEEK(occurred_at) AS week,
   device,
   COUNT(DISTINCT user_id) AS user_engagement
FROM
   events
GROUP BY 2,1
ORDER BY 1;
```

week	device	user_engagement
17	acer aspire desktop	12
17	acer aspire notebook	23
17	amazon fire phone	4
17	asus chromebook	23
17	dell inspiron desktop	20
17	dell inspiron notebook	48
17	hp pavilion desktop	17
17	htc one	19
17	ipad air	29
17	ipad mini	19
17	iphone 4s	27
17	iphone 5	69
17	iphone 5s	47
17	kindle fire	6
17	lenovo thinkpad	94

5. **Calculate the email engagement metrics?**

```
SELECT
   WEEK(occurred_at) AS week,
   COUNT(DISTINCT (CASE
         WHEN action = 'sent_weekly_digest' THEN user_id
      END)) AS weekly_digest,
   COUNT(DISTINCT (CASE
         WHEN action = 'sent_reengagement_email' THEN user_id
      END)) AS reengagement_mail,
   COUNT(DISTINCT (CASE
         WHEN action = 'email_open' THEN user_id
      END)) AS opened_email,
   COUNT(DISTINCT (CASE
         WHEN action = 'email_clickthrough' THEN user_id
      END)) AS email_clickthrough
FROM
```

```
    email_events
GROUP BY WEEK(occurred_at)
ORDER BY WEEK(occurred_at);
```

week	weekly_digest	reengagement_mail	opened_email	email_clickthrough
17	908	73	310	166
18	2602	157	900	425
19	2665	173	961	476
20	2733	191	989	501
21	2822	164	996	436
22	2911	192	965	478
23	3003	197	1057	529
24	3105	226	1136	549
25	3207	196	1084	524
26	3302	219	1149	550
27	3399	213	1207	613
28	3499	213	1228	594
29	3592	213	1201	583
30	3706	231	1363	625
31	3793	222	1338	444
32	3897	200	1318	416
33	4012	264	1417	490
34	4111	261	1502	481

Tech Stack Used

- MySQL Workbench (Version 8.0 CE): MySQL Workbench is a comprehensive software tool that facilitates tasks such as data modelling, SQL development, and administration configuration. It provides a user-friendly graphical interface, allowing users to interact with databases in a structured manner. Utilizing MySQL Workbench, you can effortlessly create databases and perform analysis to address the specific inquiries outlined in the description. Furthermore, MySQL Workbench is both user-friendly and available at no cost.
- Microsoft Excel: Used to understand the data/table provided and format the data as per the requirement.
- Microsoft Word: Used to generate a comprehensive project report that can be presented to various teams, providing a detailed overview of the project.

Insights

Case Study 1 (Operation Analytics)

- On date 2020-11-28 there is maximum number of jobs reviewed that is 218.
- On date 2020-11-28 the throughput is highest 0.06.
  Metrics will always go up and down on a weekly and daily basis. You'll get numbers faster every day or
  minute if you 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 the Persian language is the most (37.5%).
- Actor ID 1004 has duplicate rows.

Case Study 2 (Investigating Metric Spike)

- The peak user engagement of 1706 occurred in Week 30, while the lowest user engagement of 118 was recorded Week 35. As from week 17 the engagement of users was rapidly increasing till week 34 but suddenly the user engagement dropped. This means that some of the users do not find much quality in the product/service in the last of the weeks.
- There are in total 9381 active users from 1st week of 2013 to the 35th week of 2014.
- The overall count of weekly engagement per device used is the most for MacBook users and iPhone users.
- The email opening rate is around 34% and email clicking rate is around 15%. The users are engaging with the email service which is good for the company to expand.

Result

How this project helped me: This project provided me with a deep understanding of the significance of operational analytics. I gained insights into how companies leverage metric spikes as a strategic advantage. By adopting an informed and proactive approach, organizations can make data-driven decisions that optimize their strategies and drive higher return on investment (ROI).

Challenges encountered in this project: One of the challenges I faced was working with a large volume of data in Case Study 2. Due to the sheer size of the data, importing it into SQL Workbench was slow. To address this issue, I used https://tableconvert.com/ website, which facilitated the conversion of the CSV table data into SQL queries.

Conclusion: Operational analytics plays a vital role in synchronizing real-time data to drive actionable insights. By aggregating data from various sources into a cohesive and organized solution, operational analytics enables companies to create individual customer profiles and gain a holistic view of their operations. This ensures that operational processes and systems are utilized effectively. When implemented effectively, operational analytics can have a significant positive impact on society and improve efficiency in various domains.

SQL Query Link

- **Case Study 1 (Job Data): Operations Analytics**
- **Case Study 2 (Investigating Metric Spikes): Investigating Metric Spikes**