trainity

Operation Analytics & Investigating metric spike case study

Trainity

Operation Analytics and Investigating Metric Spike

- Abhishek Shukla

trainity

Operation Analytics & Investagating metric spike case study

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 are 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.

- **Abhishek Shukla**

trainity

Operation Analytics & Investigating metric spike case study

Approach

The required information was determined via SQL queries where the data base was created first in SQL and moreover for the second case study due to the size of the data excel was used to make charts for better visualisation.

- **Abhishek Shukla**

trainity

Operation Analytics & Investigating metric spike case study

Tech Stack Used

1. MySQL Workbench 8.CE was used to run the queries.
2. MS Excel was utilized for better visualization in the second case study.

- Abhishek Shukla

trainity

Operation Analytics & Investigating metric spike case study

Case Study 1 (Job Data): 1.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?

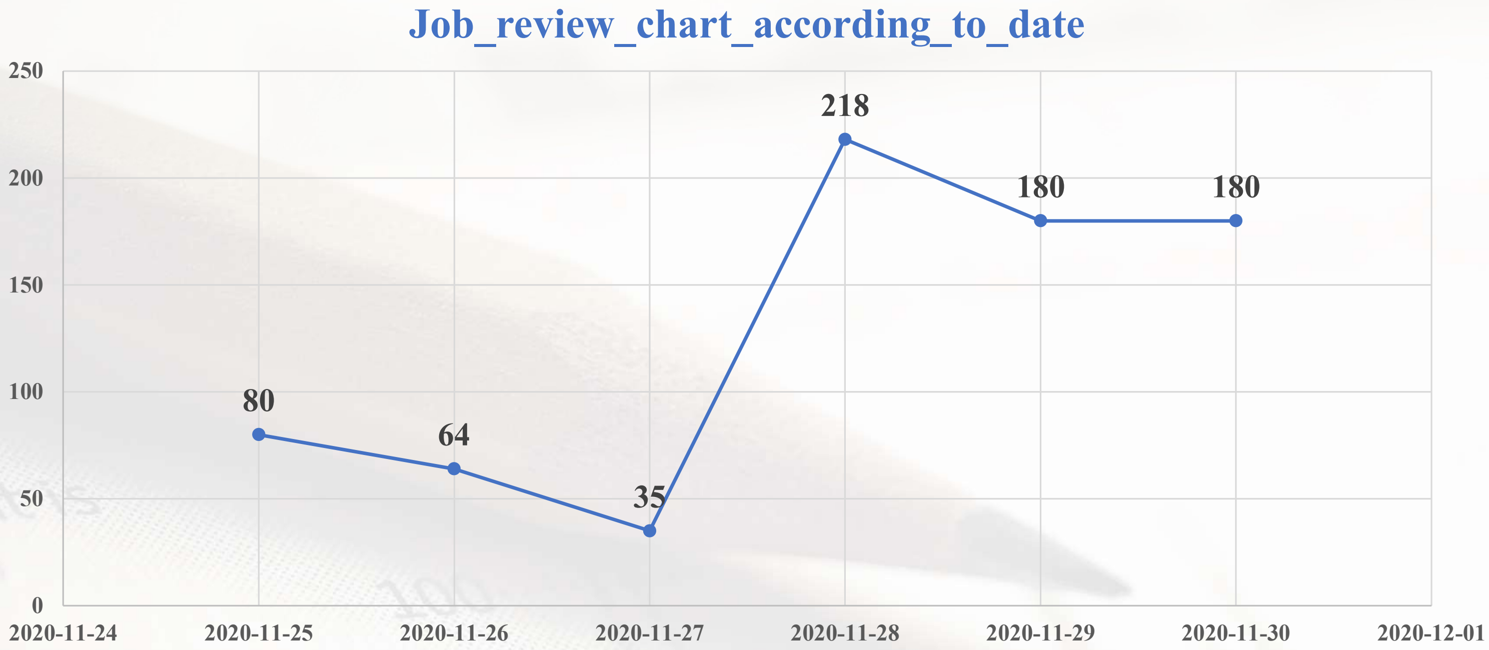
- **Abhishek Shukla**

trainity

Operation Analytics & Investagating metric spike case study

Case Study 1 (Job Data): 1.A

Trainity



Job_review_chart_according_to_date

Dates	Job_review_time
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35
2020-11-26	64
2020-11-25	80

SQL Query:

```
SELECT ds AS dates,
ROUND((COUNT(job_id)/SUM(time_
spent))*3600) AS
“Job_reviews_time” FROM job_data
WHERE ds BETWEEN ‘2020-11-01’
AND ‘2020-11-30’ GROUP BY ds;
```

Insights:

According to the task analysis, we found that the number of job reviews per hour per day in November 2020 was 757. The maximum number of job reviews occurred on November 30th and 29th, 2020, with a total of 180 reviews. On November 27th, 2020, the minimum number of job reviews was 35.

trainity

Operation Analytics & Investagating metric spike case study

Case Study 1 (Job Data): 1.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?

- **Abhishek Shukla**

trainity

Operation Analytics & Investagating metric spike case study

Case Study 1 (Job Data): 1.B

Trainity

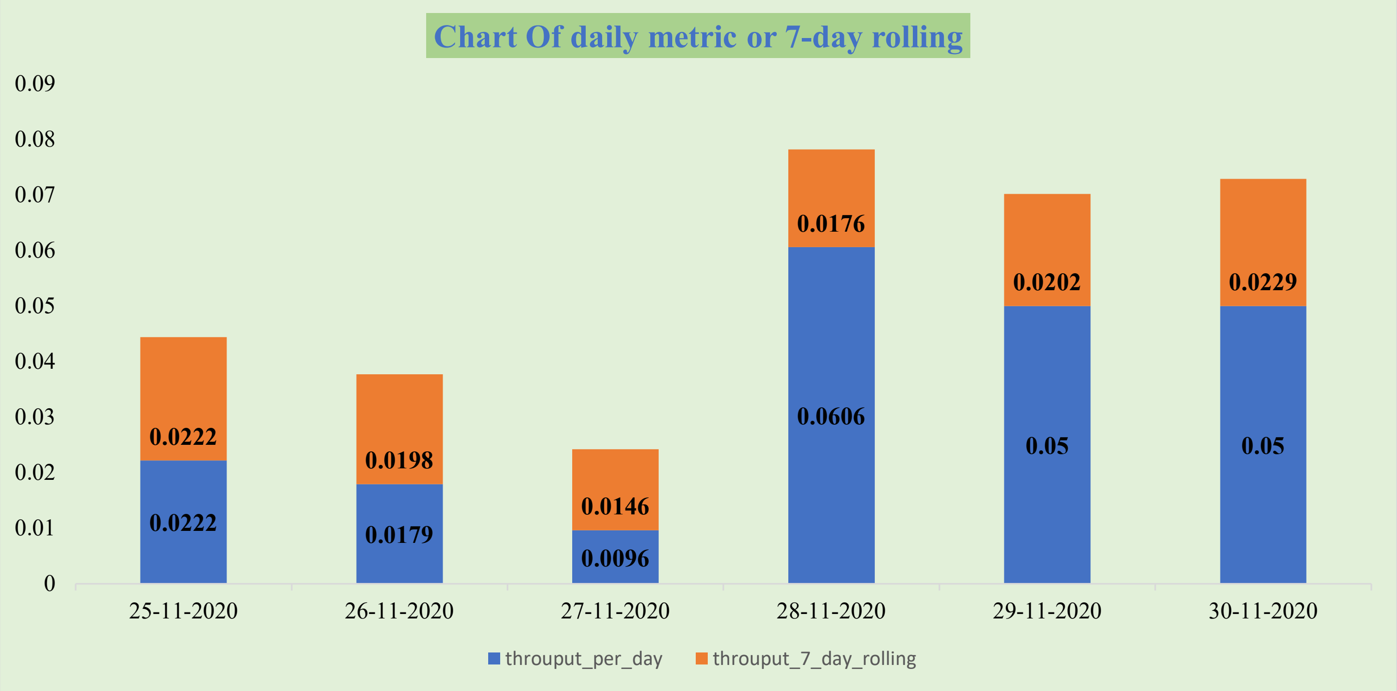


Chart Of daily metric or 7-day rolling

ds	throuput_per_day	throuput_7_day_ rolling
25-11-2020	0.0222	0.0222
26-11-2020	0.0179	0.0198
27-11-2020	0.0096	0.0146
28-11-2020	0.0606	0.0176
29-11-2020	0.05	0.0202
30-11-2020	0.05	0.0229

SQL Query:

```
Select ds, c/t as throuput_per_day, c7/s7 as
throuput_7_day_rolling
From (select ds, count(job_id) as c, sum(time_spent) as t,
count(job_id) over(order by ds rows between  6 preceding  and
current row) as c7, sum(time_spent) over(order by ds rows
between 6 preceding  and current row) as s7 from job_data
where  month(ds)=11
group by ds) a;
```

Insights:

The 7-day rolling average is better because it can offset the fluctuations in throughput from one day to another, creating a more accurate picture. In this context, the expression "c/t" represents the calculation of throughput per day. Let's break it down:

"c" refers to the count of job reviews for a particular day.

"t" refers to the total time spent on job reviews for that same day.

By dividing "c" by "t," we obtain the average number of job reviews per unit of time, which in this case is per day. This ratio provides an indication of the efficiency or productivity in terms of job reviews completed within a specific timeframe.

trainity

Operation Analytics & Investigating metric spike case study

Case Study 1 (Job Data): 1.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?

- **Abhishek Shukla**

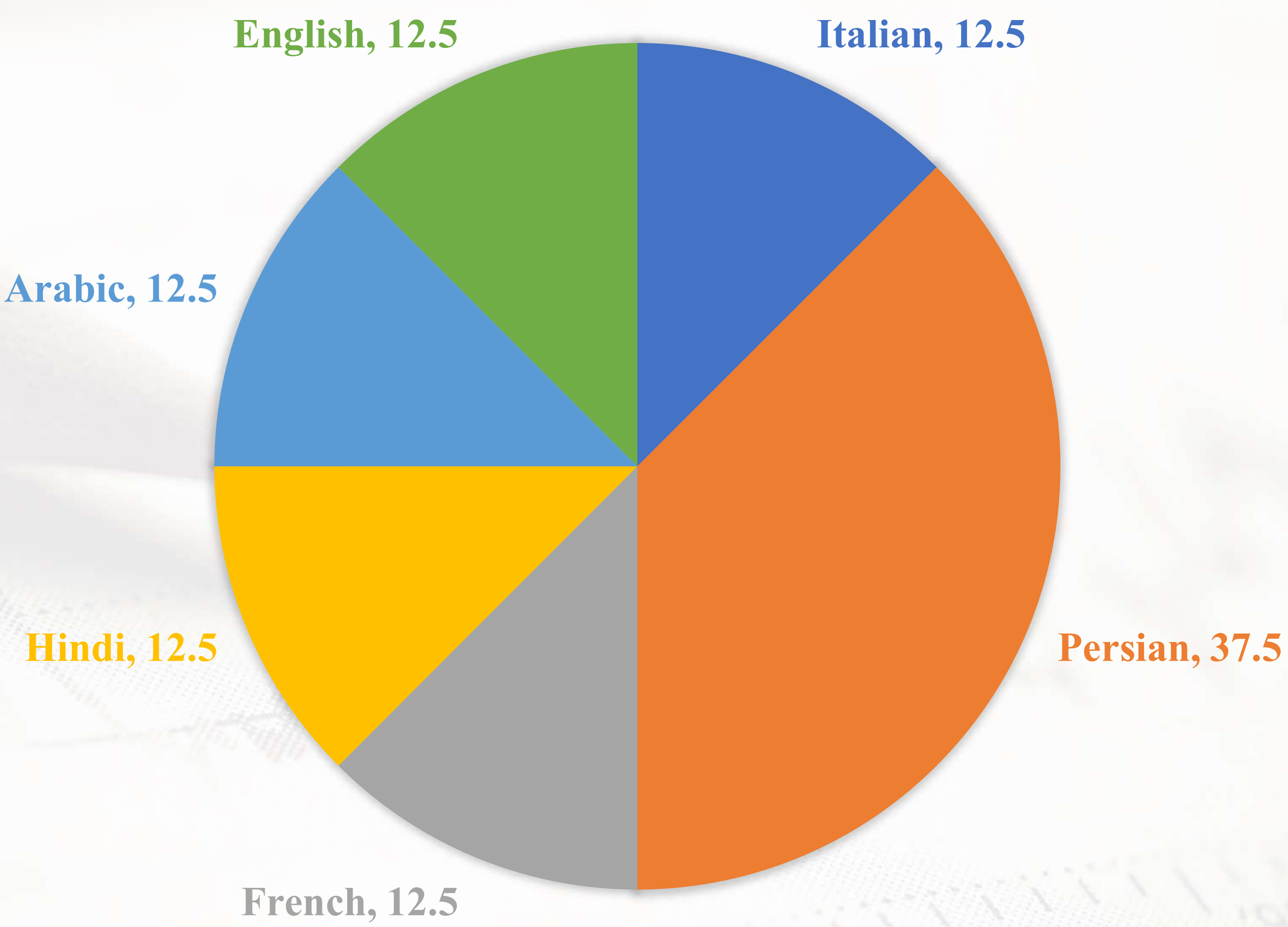
Operation Analytics & Investigating metric spike case study

Case Study 1 (Job Data): 1.C

Trainity

SQL Query:

```
with a as
(select max(ds)  as m from
job_data)
select distinct language,
(count(event) over(partition
by language rows between
unbounded preceding and
unbounded following)
/count(*) over(order by ds
rows between unbounded
preceding and unbounded
following) ) * 100 as
percentage
 from (select * From job_data
cross join a
Where datediff(m,date(ds))
between 0 and 30)a1;
```



PERCENTAGE SHARE OF EACH LANGUAGE

trainity

Operation Analytics & Investigating metric spike case study

Case Study 1 (Job Data): 1.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?

- **Abhishek Shukla**

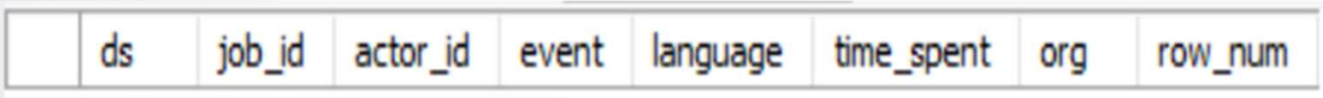
When no duplicate data

Case Study 1 (Job Data): 1.D

Trainity

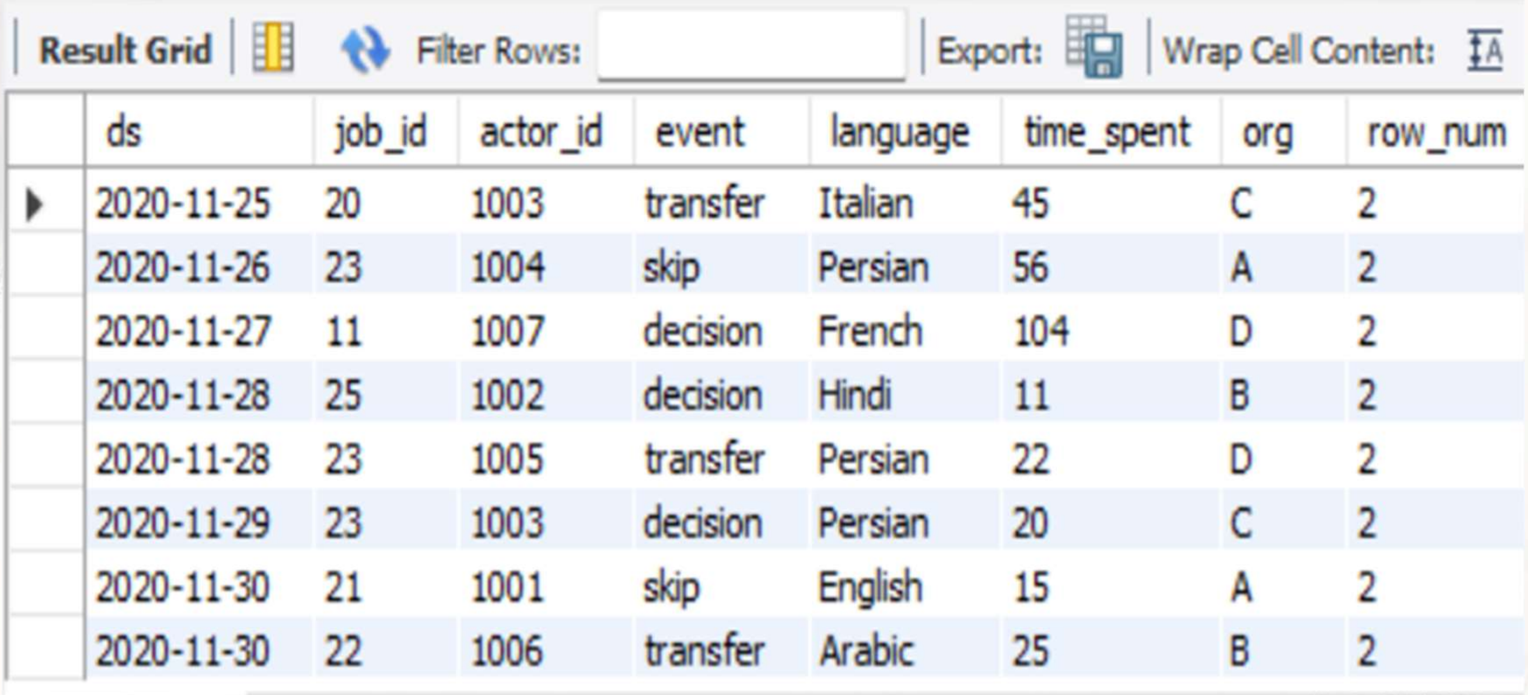
SQL Query:

```
select * from( select *,
row_number() over(partition by
ds,actor_id,job_id) as row_num
From  job_data) a
where row_num>1;
```



	ds	job_id	actor_id	event	language	time_spent	org	row_num

When duplicates(inserted the same data twice for the example)



	ds	job_id	actor_id	event	language	time_spent	org	row_num
▶	2020-11-25	20	1003	transfer	Italian	45	C	2
	2020-11-26	23	1004	skip	Persian	56	A	2
	2020-11-27	11	1007	decision	French	104	D	2
	2020-11-28	25	1002	decision	Hindi	11	B	2
	2020-11-28	23	1005	transfer	Persian	22	D	2
	2020-11-29	23	1003	decision	Persian	20	C	2
	2020-11-30	21	1001	skip	English	15	A	2
	2020-11-30	22	1006	transfer	Arabic	25	B	2

trainity

Operation Analytics & Investagating metric spike case study

Trainity

Case Study- 2: Investigating Metric Spike

Insights

- Abhishek Shukla

trainity

Operation Analytics & Investagating metric spike case study

Case Study 2 (Investigating metric spike): 2.A

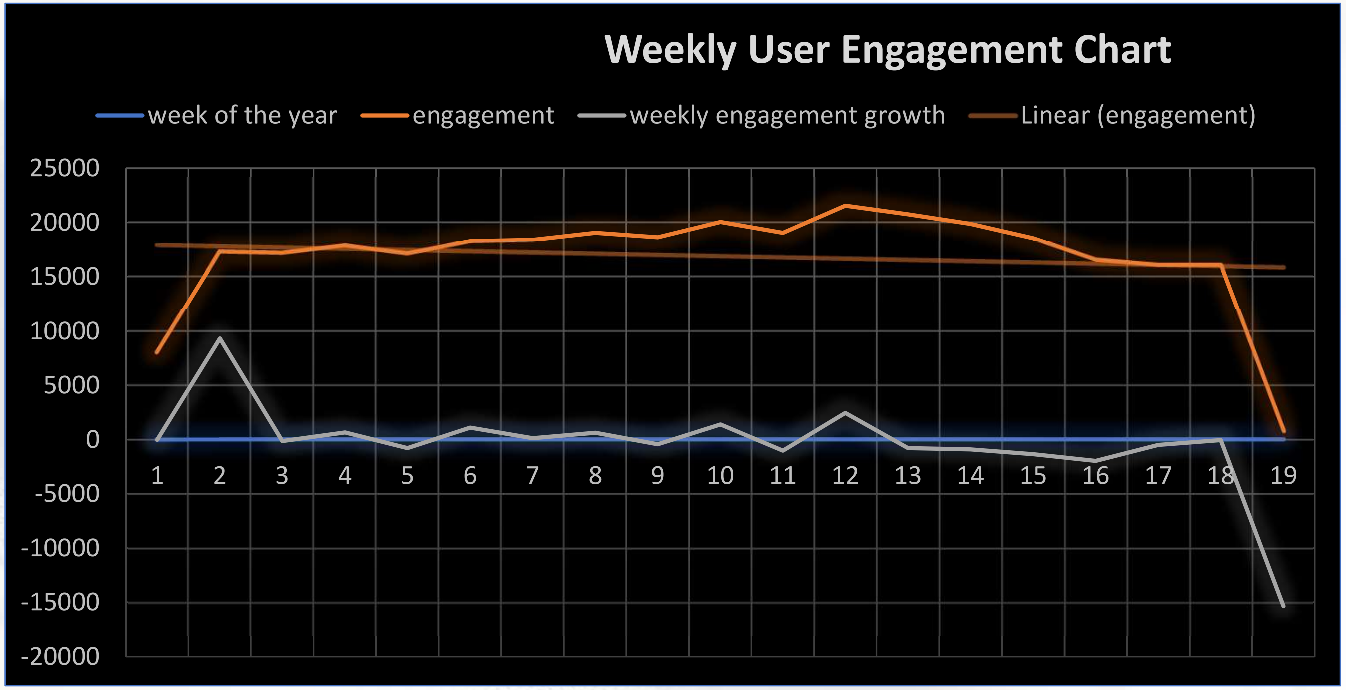
User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service.

Your task: Calculate the weekly user engagement?

- **Abhishek Shukla**

Case Study 2 (Investigating metric spike) 2.A

Trainity



week of the year	engagement	weekly engagement growth
17	8019	NULL
18	17341	9322
19	17224	-117
20	17911	687
21	17151	-760
23	18280	1129
22	18413	133
24	19052	639
25	18642	-410
29	20067	1425
26	19061	-1006
30	21533	2472
28	20776	-757
27	19881	-895
31	18556	-1325
32	16612	-1944
33	16145	-467
34	16127	-18
35	784	-15343

SQL Query:

```
select *, engagement-lag(engagement) over(partition by'week
of the year') as 'weekly engagement growth’
From (select  week(occurred_at) as 'week of the year’,
count(event_name) as 'engagement’
from events
where event_type!='signup_flow’
group by week(occurred_at))a;
```

Insights:

An overall reduction in engagement is observed. (*Note: The data for the 35th should not be considered as it represents only the first day of the week.)

trainity

Operation Analytics & Investigating metric spike case study

Case Study 2 (Investigating metric spike): 2.B

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

Your task: Calculate the user growth for product?

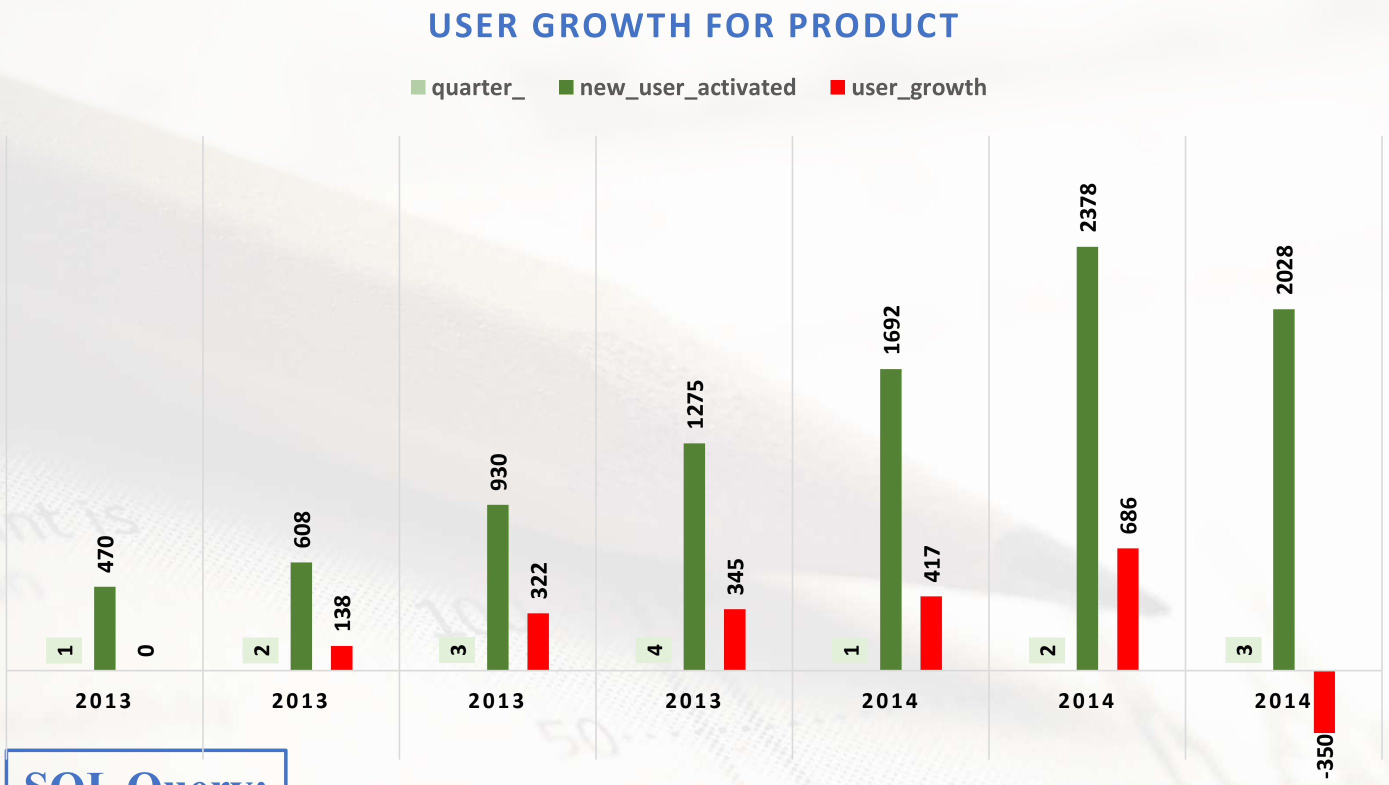
- **Abhishek Shukla**

trainity

Operation Analytics & Investigating metric spike case study

Case Study 2 (Investigating metric spike) 2.B

Trainity



year_	quarter_	new_user_activated	user_growth
2013	1	470	NULL
2013	2	608	138
2013	3	930	322
2013	4	1275	345
2014	1	1692	417
2014	2	2378	686
2014	3	2028	-350

SQL Query:

```
select *, new_user_activated-lag(new_user_activated) over( order by
year_,quarter_ ) as user_growth
from(select year(created_at) as year_,quarter(created_at) as
quarter_,count(user_id) as new_user_activated
from users
where activated_at is not null and state='active’
group by 1,2)a ;
```

Insights:

An overall increase in quarterly performance is evident. (*Please note that the data for the third quarter of 2014 does not represent the full quarter.)

trainity

Operation Analytics & Investigating metric spike case study

Case Study 2 (Investigating metric spike): 2.C

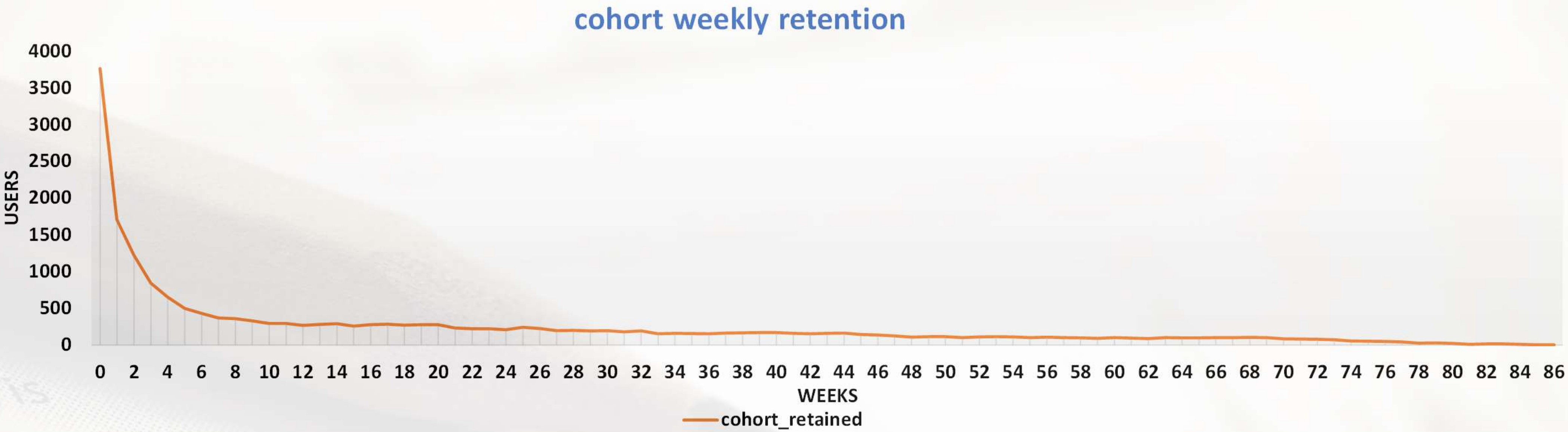
Weekly Retention: Users getting retained weekly after signing-up for a product.

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

- **Abhishek Shukla**

Operation Analytics & Investigating metric spike case study

Case Study 2 (Investigating metric spike) 2.C



SQL Query:

```
Select
week_period, first_value(cohort_retained) over (order by week_period)
as cohort_size, cohort_retained, cohort_retained /
first_value(cohort_retained) over (order by week_period) as
pct_retained  From (select
timestampdiff(week,a.activated_at,b.occurred_at) as week_period,
count(distinct a.user_id) as cohort_retained From
(select user_id, activated_at from users where state='active'group by 1)
a
inner join (select user_id,occurred_at from events )b
on a.user_id=b.user_id group by 1) c;
```

Insights:

There was a significant drop in the first 10 weeks, and by the end of 85 weeks, only 2 users remained.

Operation Analytics & Investigating metric spike case study

Case Study 2 (Investigating metric spike): 2.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?

- **Abhishek Shukla**

Operation Analytics & Investigating metric spike case study

Case Study 2 (Investigating metric spike) 2.D

Trainity

SQL Query:

```
Select
device_name,
avg(num_users_using_device) as avg_weekly_users,
avg(times_device_use_current_week) as
avg_times_used_weekly
From
(select week(occurred_at) as week,
device as device_name ,
count(distinct user_id) as num_users_using_device,
count(device) as times_device_use_current_week
from events
where event_name='login’
group by 1,2
order by 1) a
group by 1;
```

Insights:

The average weekly engagement per device was calculated based on a large dataset (960 rows). It was found that Macbook Pro was the most commonly used device, while Samsung Galaxy Tablet was the least used device.

device_name	avg_weekly_users	avg_times_used_weekly
acer aspire desktop	26	32.9474
acer aspire notebook	43.1579	56.8421
amazon fire phone	10.5556	13.7778
asus chromebook	43.5263	58.8947
dell inspiron desktop	46.6316	62.7368
dell inspiron notebook	91.1053	123.4737
hp pavilion desktop	42.1053	55.8421
htc one	21.8421	27.6842
ipad air	51.4444	61.7222
ipad mini	30	34.7368
iphone 4s	46.6316	60.5789
iphone 5	123.1579	161.2105
iphone 5s	73.3158	96.7895
kindle fire	21.1579	25.5263
lenovo thinkpad	172.9474	232.5789
mac mini	20.4737	27.3684
macbook air	123.1579	164.8947
macbook pro	260.1579	358.1579
nexus 10	27.0526	31.8421
nexus 5	76.3684	99.6316
nexus 7	36.3684	43.2632
nokia lumia 635	28.1579	36.2632
samsumg galaxy tablet	10.2778	12.1111
samsung galaxy note	13.4737	17.5789
samsung galaxy s4	91.5789	118.7368
windows surface	18.2105	21.5263

Operation Analytics & Investigating metric spike case study

Case Study 2 (Investigating metric spike): 2.E

Email Engagement: Users engaging with the email service.

Your task: Calculate the email engagement metrics?

- **Abhishek Shukla**

trainity

Operation Analytics & Investigating metric spike case study

Case Study 2 (Investigating metric spike) 2.E

Trainity

week	num_users	time_weekly_digest_sent	time_weekly_digest_sent_growth	time_email_open	time_email_open_growth	time_email_clickthrough	time_email_clickthrough_growth
17	981	908	NULL	310	NULL	166	NULL
18	2714	2602	1694	912	602	430	264
19	2787	2665	63	972	60	477	47
20	2874	2733	68	1004	32	507	30
21	2926	2822	89	1014	10	443	-64
22	3029	2911	89	987	-27	488	45
23	3134	3003	92	1075	88	538	50
24	3254	3105	102	1155	80	554	16
25	3343	3207	102	1096	-59	530	-24
26	3439	3302	95	1165	69	556	26
27	3543	3399	97	1228	63	621	65
28	3641	3499	100	1250	22	599	-22
29	3734	3592	93	1219	-31	590	-9
30	3866	3706	114	1383	164	630	40
31	3950	3793	87	1351	-32	445	-185
32	4023	3897	104	1337	-14	418	-27
33	4200	4012	115	1432	95	490	72
34	4294	4111	99	1528	96	490	0
35	48	0	-4111	41	-1487	38	-452

SQL Query:

```
Select week, num_users, time_weekly_digest_sent,
time_weekly_digest_sent-lag(time_weekly_digest_sent)
over(order by week) as time_weekly_digest_sent_growth,
time_email_open,time_email_open-lag(time_email_open)
over(order by week) as time_email_open_growth,
time_email_clickthrough,time_email_clickthrough-
lag(time_email_clickthrough) over(order by week) as
time_email_clickthrough_growth
From (select week(occurred_at)as week, count(distinct user_id)
as num_users, sum(if(action='sent_weekly_digest',1,0)) as
time_weekly_digest_sent, sum(if(action='email_open',1,0)) as
time_email_open, sum(if(action='email_clickthrough',1,0)) as
time_email_clickthrough
from email
group by 1
order by 1) a;
```

Result:

This project was truly engaging, and the level of difficulty made it even more fulfilling to execute. I learned a lot of new concepts, such as rolling averages and cohort retention analysis. I made an effort to include Excel charts wherever possible, and I hope to improve my efficiency in using Excel for future projects. Additionally, I have become more proficient in using Windows functions.

- **Abhishek Shukla**