Operation Analytics and Investigating Metric Spike

Shahnawaz Akhtar
shahnawazakhtar2@gmail.com

Project Description

The provided project includes two case studies.

- The first task pertains to Operation Analytics, involving the analysis of job data. This includes identifying the total number of jobs reviewed, calculating the 7-day rolling average of throughput, determining the percentage share of different languages used, and identifying any duplicate entries.
- The second task involves examining a spike in metrics, focusing on analyzing user engagement, user growth, weekly retention rates, weekly engagement levels, and email engagement statistics.

Approach

The necessary information was obtained using MySQL Workbench, starting with the creation of a database in SQL. Additionally, for the second case study, Excel was utilized to create charts for better visualization due to the large size of the data.

Tech Stack Used

- The SQL queries were executed using MySQL workbench 8.0 community server version 8.0.36 which is owned by oracle.
- MS Excel was used in the second case study for better visualisation.

Case Study-1

JOB DATA ANALYSIS

Task-A: Jobs Reviewed Over Time-

Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

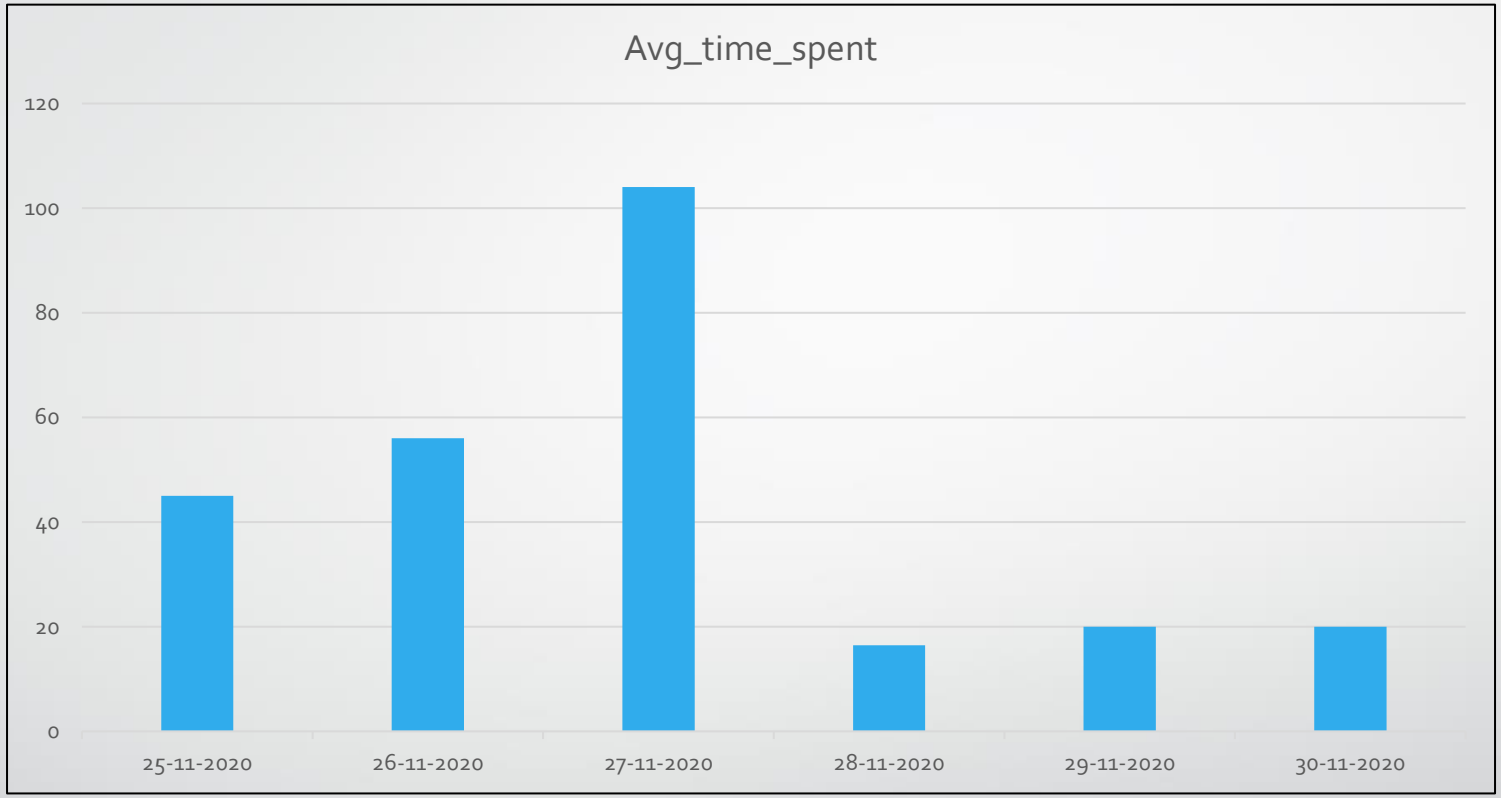
Query:

```
SELECT ds, SUM(time_spent)/COUNT(*) AS avg_time_spent
FROM job_data
WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY ds
ORDER BY ds;
```

Output:

Ds	Avg_time_spent
2020-11-25	45.0000
2020-11-26	56.0000
2020-11-27	104.0000
2020-11-28	16.5000
2020-11-29	20.0000
2020-11-30	20.0000

Task-A: Visualization



INSIGHTS:
The maximum 104 jobs reviewed per on 27 Nov, 2020 and minimum 16.50 jobs reviewed per hour on 28 Nov, 2020

Task-B: Throughput Analysis:

Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

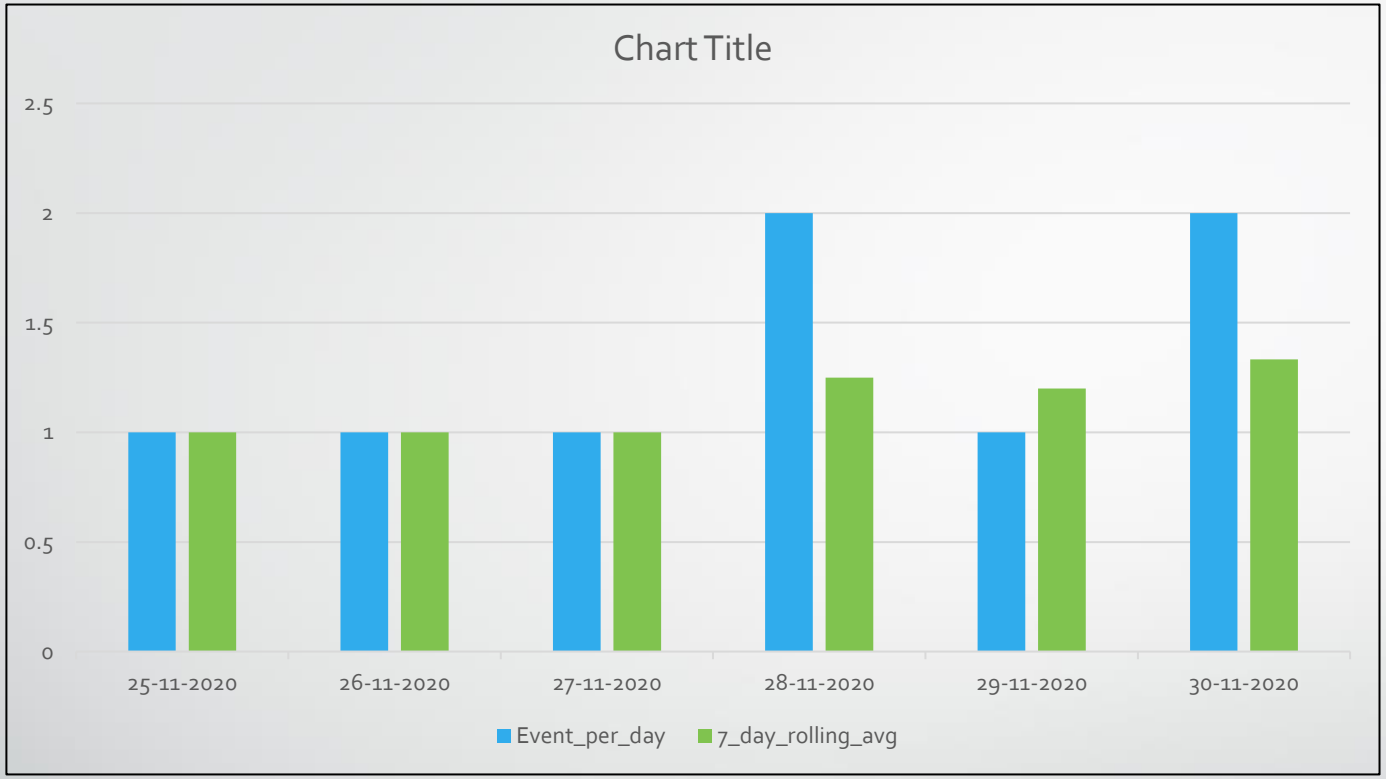
Query:

```
SELECT ds, event_per_day, AVG(event_per_day) OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND
CURRENT ROW) AS 7_day_rolling_avg
FROM (SELECT ds, COUNT(DISTINCT event) AS event_per_day
FROM job_data
WHERE ds BETWEEN '2020-11-01' and '2020-11-30'
GROUP BY ds
ORDER BY ds)a;
```

Output:

ds	Event_per_day	7_day_rolling_avg
2020-11-25	1	1.0000
2020-11-26	1	1.0000
2020-11-27	1	1.0000
2020-11-28	2	1.2500
2020-11-29	1	1.2000
2020-11-30	2	1.3333

Task-B: Visualization



Insights:

The 7-day rolling average of throughput offers a smooth perspective on the data, helping you track trends over time while minimizing the impact of daily variations.

Task-C: Language Share Analysis:

Write an SQL query to calculate the percentage share of each language over the last 30 days.

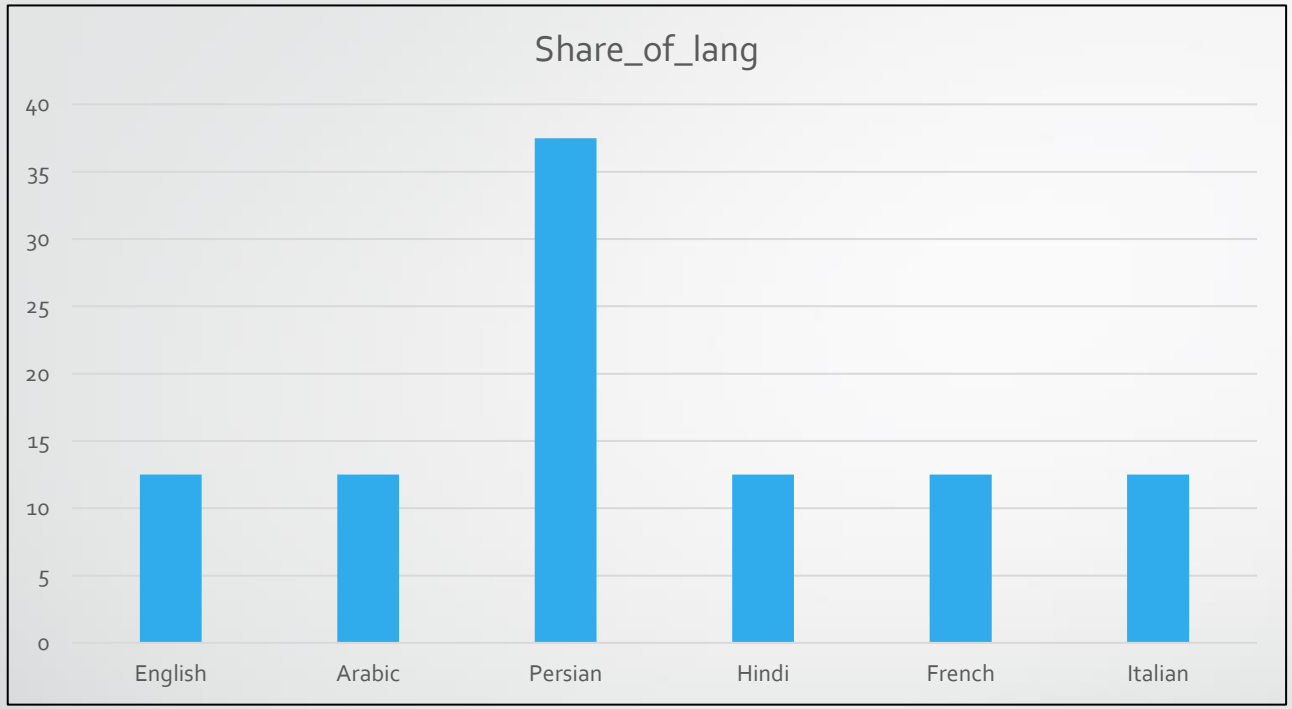
Query:

```
select language, round(((count(language)/8)*100),2) as share_of_lang
from job_data
group by language;
```

Output:

Language	Share_of_lang
English	12.50
Arabic	12.50
Persian	37.50
Hindi	12.50
French	12.50
Italian	12.50

Task-C: Visualization



Insights:

The language distribution over the past 30 days is quite balanced, with Persian standing out as the language with the highest proportion.

Task-D: Duplicate Rows Detection:

Write an SQL query to display duplicate rows from the job_data table.

Query:

```
SELECT *
FROM job_data
WHERE job_id IN
(SELECT job_id FROM job_data GROUP BY job_id HAVING COUNT(*) > 1)
```

Output:

Ds	Job_id	Actor_id	Event	Language	Time_spent	org
2020-11-29	23	1003	decision	Persian	20	C
2020-11-28	23	1005	transfer	Persian	22	D
2020-11-26	23	1004	skip	Persian	56	A

Insights:

The query identifies that there are 3 job IDs in the job_data table with duplicate entries. This means that 3 different job IDs appear more than once in the table.

Case Study-2

INVESTIGATING METRIC SPIKE

Task-A: Weekly User Engagement:

Write an SQL query to calculate the weekly user engagement.

Query:

```
select extract(week from occurred_at) as week_number, count(distinct
user_id) as active_user
from events
where event_type='engagement'
group by week_number
order by week_number;
```

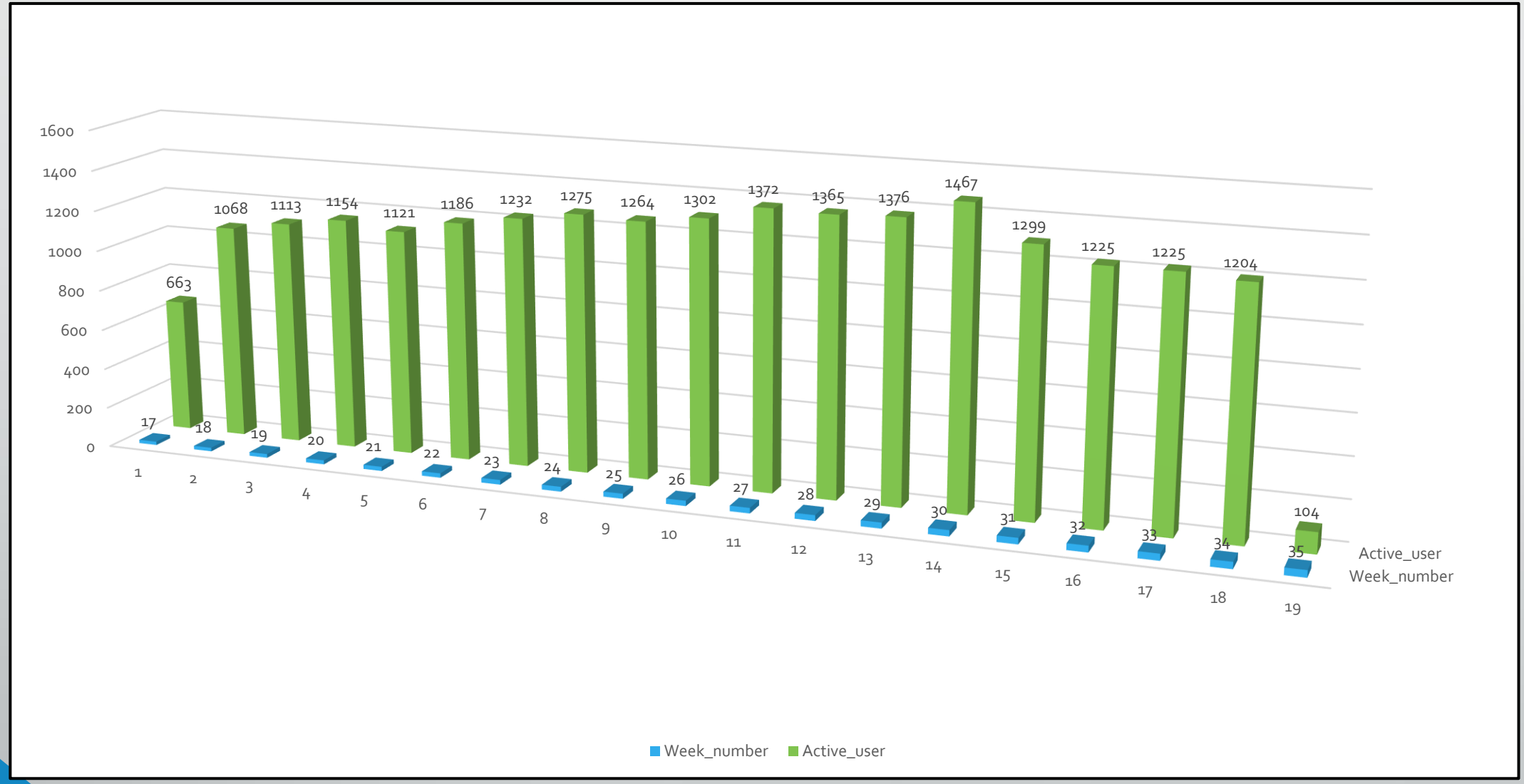
Output:

Week_number	Active_user
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

INSIGHTS:

In week 30, there was the highest number of user engagements, while week 35 saw the lowest number of user engagements.

Task-A: Visulalization



Task-B: User Growth Analysis:

Write an SQL query to calculate the user growth for the product.

Query:

```
select  year_num, week_num, num_active_users,
sum(num_active_users) over (order by year_num, week_num
rows between unbounded preceding and current row) as
cum_active_users
from (select extract(year from a.activated_at) as year_num,
extract(week from a.activated_at) as week_num,
count(distinct user_id) as num_active_users
from  users a
where state = 'active'
group by year_num, week_num
order by year_num,week_num) a;
```

INSIGHTS:

Highest no. of active users in week 52, and lowest no. of active users in week 2. Overall user growth start from week by week

Task-B: User Growth Analysis:

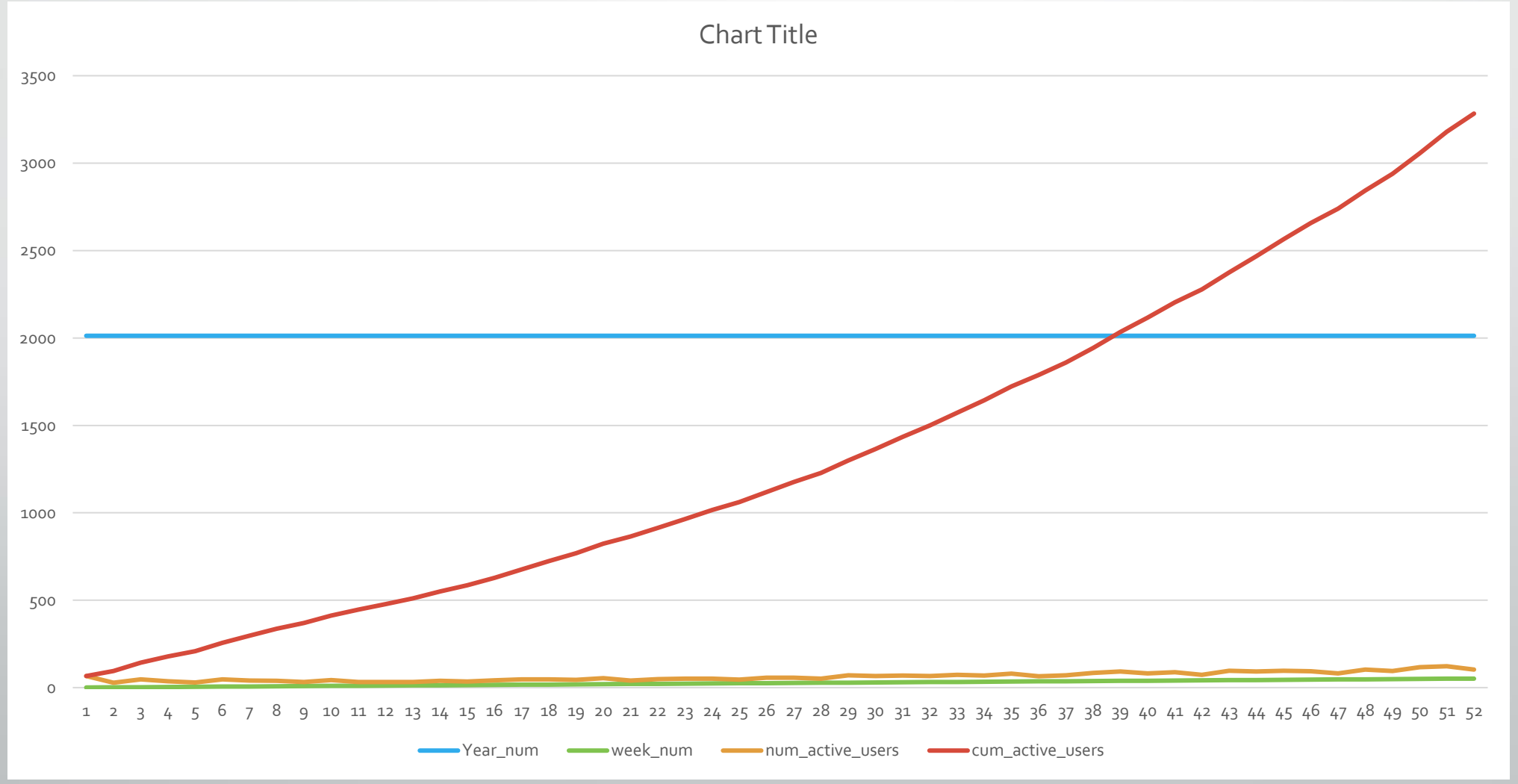
Write an SQL query to calculate the user growth for the product.

Output:

Year_num	week_num	num_active_users	cum_active_users
2013	1	67	67
2013	2	29	96
2013	3	47	143
2013	4	36	179
2013	5	30	209
2013	6	48	257
2013	7	41	298
2013	8	39	337
2013	9	33	370
2013	10	43	413
2013	11	33	446
2013	12	32	478
2013	13	33	511
2013	14	40	551
2013	15	35	586
2013	16	42	628
2013	17	48	676
2013	18	48	724
2013	19	45	769
2013	20	55	824
2013	21	41	865
2013	22	49	914
2013	23	51	965
2013	24	51	1016
2013	25	46	1062
2013	26	57	1119
2013	27	57	1176
2013	28	52	1228
2013	29	71	1299
2013	30	66	1365
2013	31	69	1434
2013	32	66	1500
2013	33	73	1573
2013	34	70	1643
2013	35	80	1723
2013	36	65	1788
2013	37	71	1859
2013	38	84	1943
2013	39	92	2035
2013	40	81	2116
2013	41	88	2204
2013	42	74	2278
2013	43	97	2375
2013	44	92	2467
2013	45	97	2564
2013	46	94	2658
2013	47	82	2740
2013	48	103	2843
2013	49	96	2939
2013	50	117	3056
2013	51	123	3179
2013	52	104	3283

Task-B: User Growth Analysis:

Visualization chart:



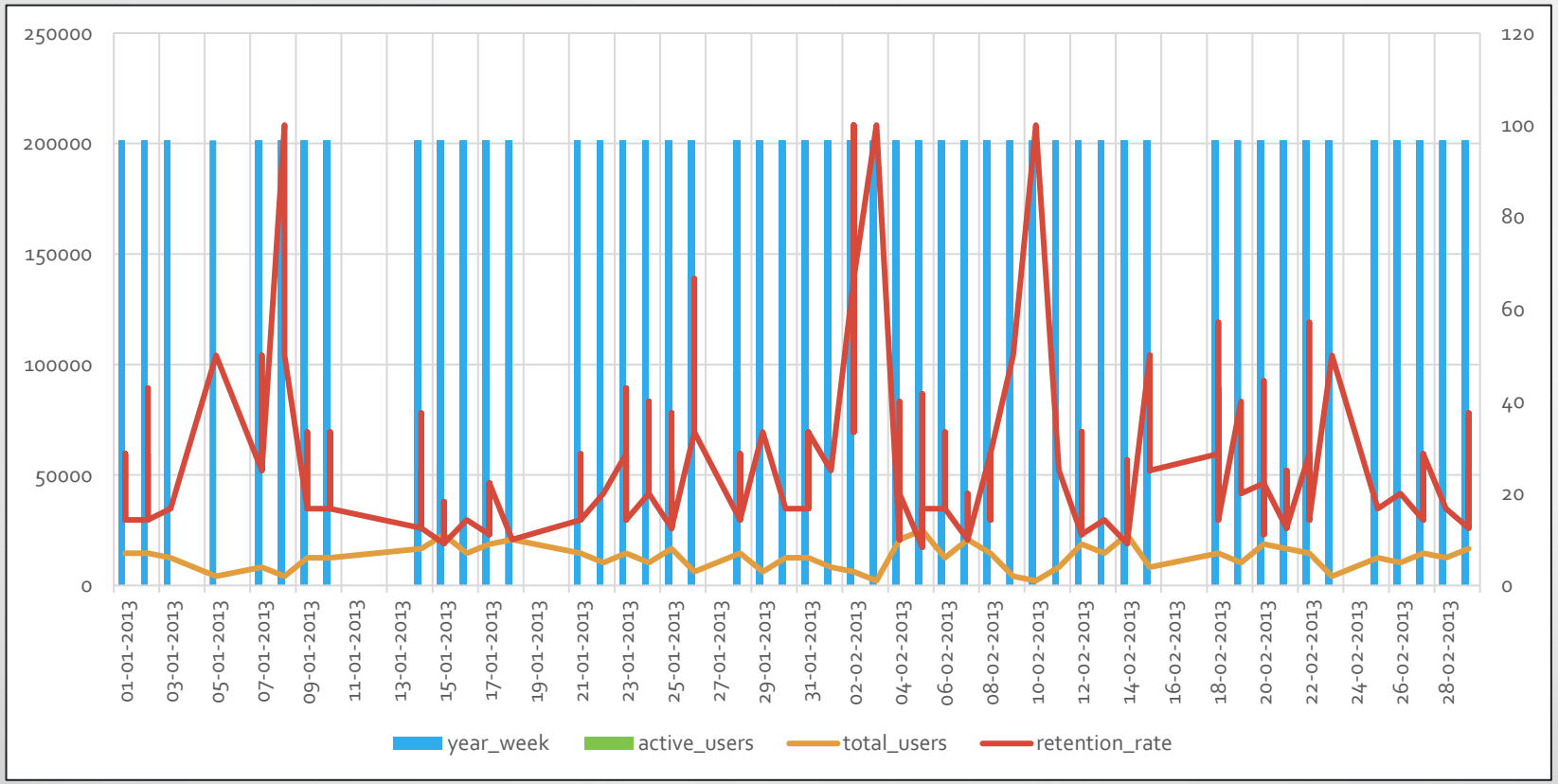
Task-C: Weekly Retention Analysis:

Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

Query:

```
with cohorts as (select date(activated_at) as cohort_start_date,
count(*) as total_users    from users    group by 1),weekly_stats as (
select       date(u.activated_at) as cohort_start_date,
yearweek(e.occurred_at, 0) as year_week,       count(distinct e.user_id)
as active_users    from users u    join events e on u.user_id = e.user_id
where e.event_type = 'engagement'    group by cohort_start_date,
year_week)select    cohorts.cohort_start_date,
weekly_stats.year_week,    weekly_stats.active_users,
cohorts.total_users as total_users,    weekly_stats.active_users /
cohorts.total_users * 100 as retention_ratefrom cohortsjoin
weekly_stats    on cohorts.cohort_start_date =
weekly_stats.cohort_start_dateorder by cohort_start_date,
year_week;
```

Visualization:



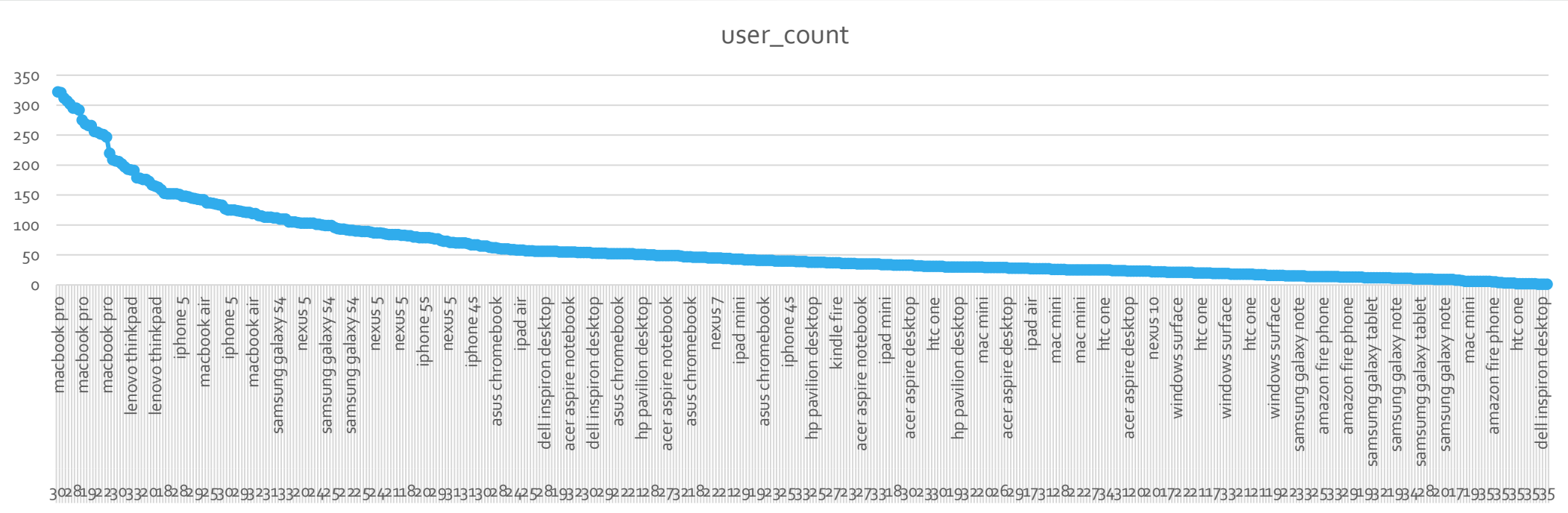
Task-D: Weekly Engagement Per Device:

Write an SQL query to calculate the weekly engagement per device

Query:

```
with cte as (select extract(week from occurred_at) as weeknum,
device, count(distinct user_id) as usercnt
from events
where event_type ='engagement'
group by weeknum, device order by weeknum)
select weeknum, device, usercnt
from cte
```

Visualization:



Given is average weekly engagement per device

The weekly data per device was very large (960 rows) hence calculated the weekly data

Macbook pro is used the most

Dell Inspiron desktop is used least

Task-E: Email Engagement Analysis:

Write an SQL query to calculate the email engagement metrics.

Query:

```
select action, extract(month from occurred_at) as month,
count(action) as number_of_mails
from email_events
group by action, month
order by action, month;
```

Email_clickthough maximum in 7th month,

email_open maximum engagement in 8th month,

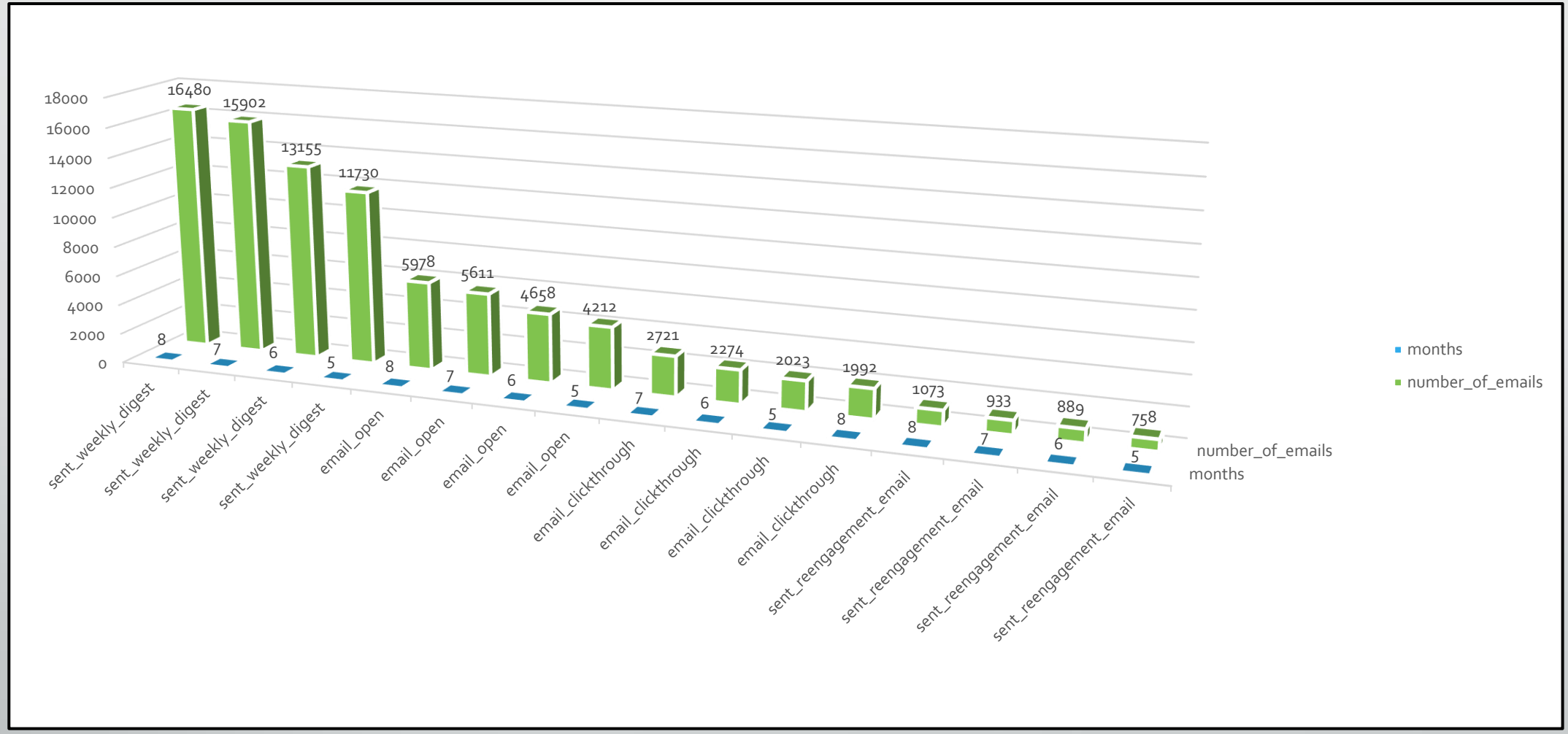
Sent_reegagment_email maximum engagement in 8th month,

Sent_weekly_digest maximum engagement in 8th month

Output

action	month	number_of_mails
email_clickthrough	5	2023
email_clickthrough	6	2274
email_clickthrough	7	2721
email_clickthrough	8	1992
email_open	5	4212
email_open	6	4658
email_open	7	5611
email_open	8	5978
sent_reengagement_email	5	758
sent_reengagement_email	6	889
sent_reengagement_email	7	933
sent_reengagement_email	8	1073
sent_weekly_digest	5	11730
sent_weekly_digest	6	13155
sent_weekly_digest	7	15902
sent_weekly_digest	8	16480

Visualization:



Result

- This project is highly engaging, and the challenges involved make it even more satisfying to complete.
- I gained new knowledge in areas such as rolling averages and cohort retention analysis.
- I tried to incorporate Excel charts wherever possible, and I hope to use Excel more efficiently in the future.
- Working on this project has allowed me to delve into the intricacies of operational analytics, gaining a deeper understanding of its principles and methodologies.
- I have learned how to effectively merge and normalize diverse datasets, ensuring accurate and reliable analysis.


Thank
you