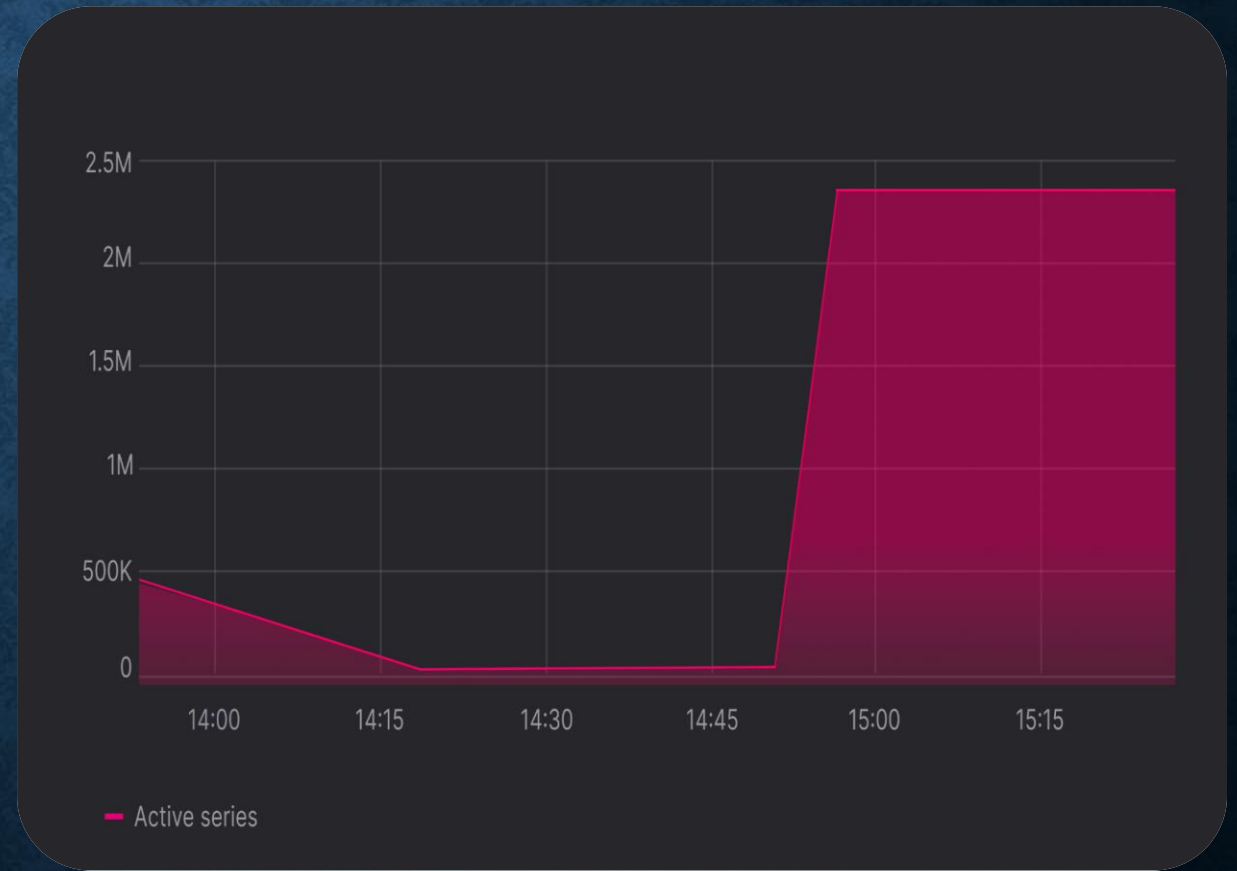


OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE



TOOLS USED



Ms -Excel



MS-PowerBI

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.

You 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.

CASE STUDY - 1

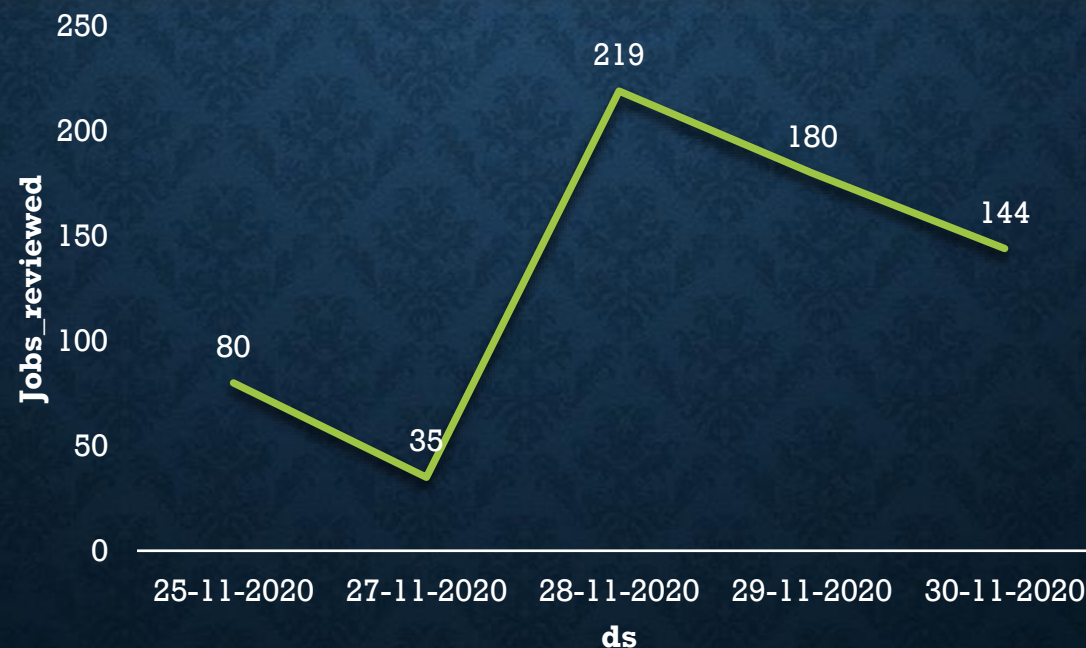
Operation-1 (Table-1)

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

1.Number of jobs reviewed: amount of jobs reviewed over time.

Task: calculate the number of jobs reviewed per hour per day for November 2020?

```
4 • select ds, ceil(count(job_id)*3600/sum(time_spent)) as jobs_reviewed      -- jobs reviewed/hour/day = count(job_id)/sum(time_spent in sec)* 1/3600 in hr
5   from oa.job_data                                                         -- and per day is given by group by ds
6   where event in('transfer','decision')
7   and ds between '1-11-2020' and '30-11-2020'
8   group by 1
9   order by 1;
```

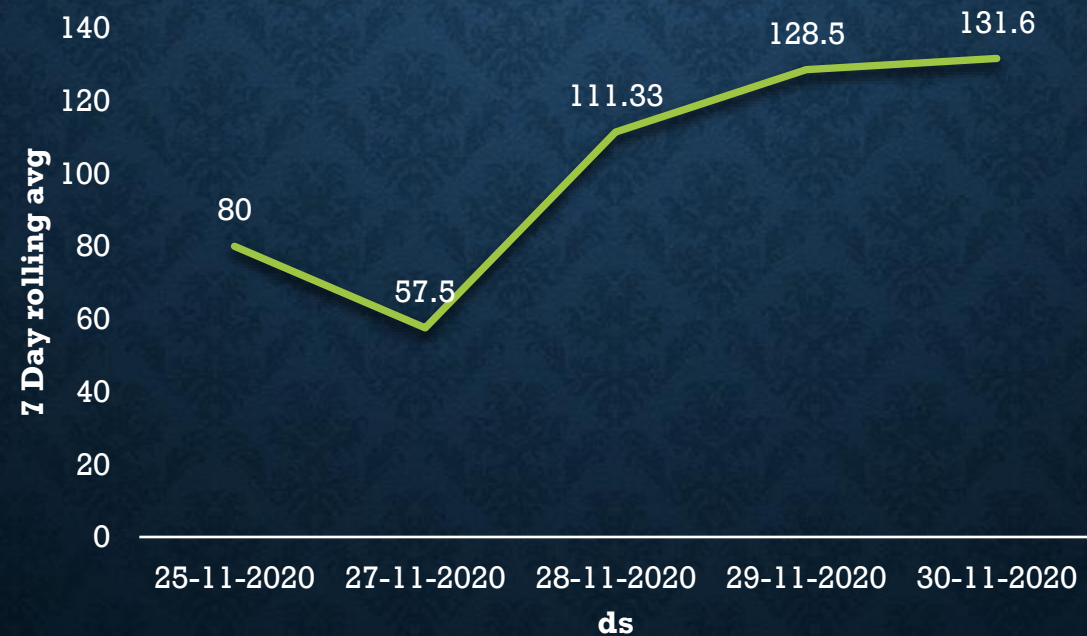


28-11-2020 have a
metric spike

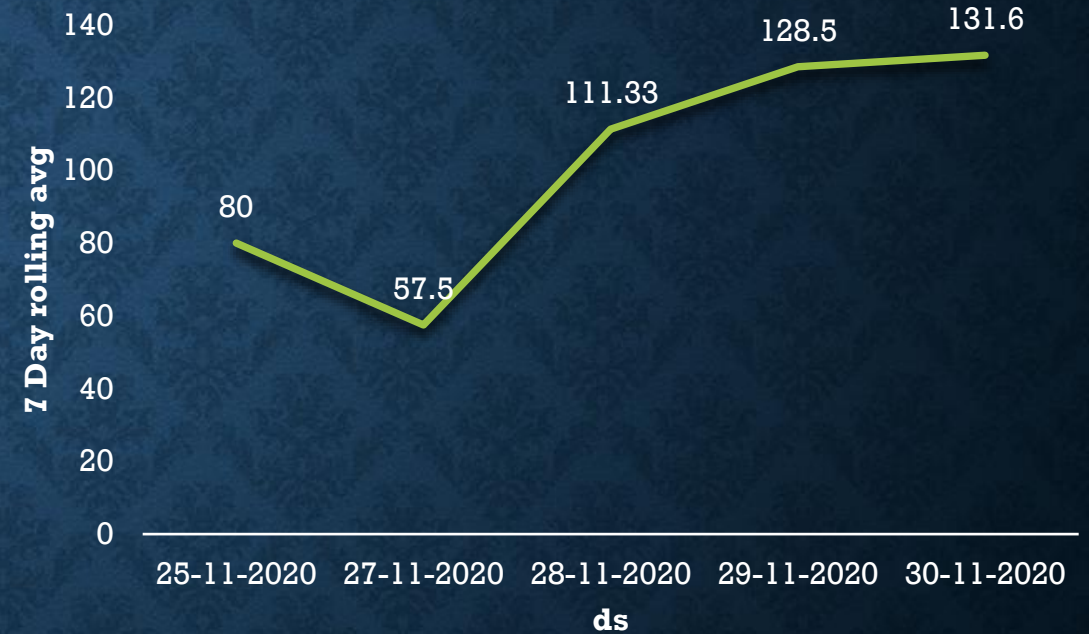
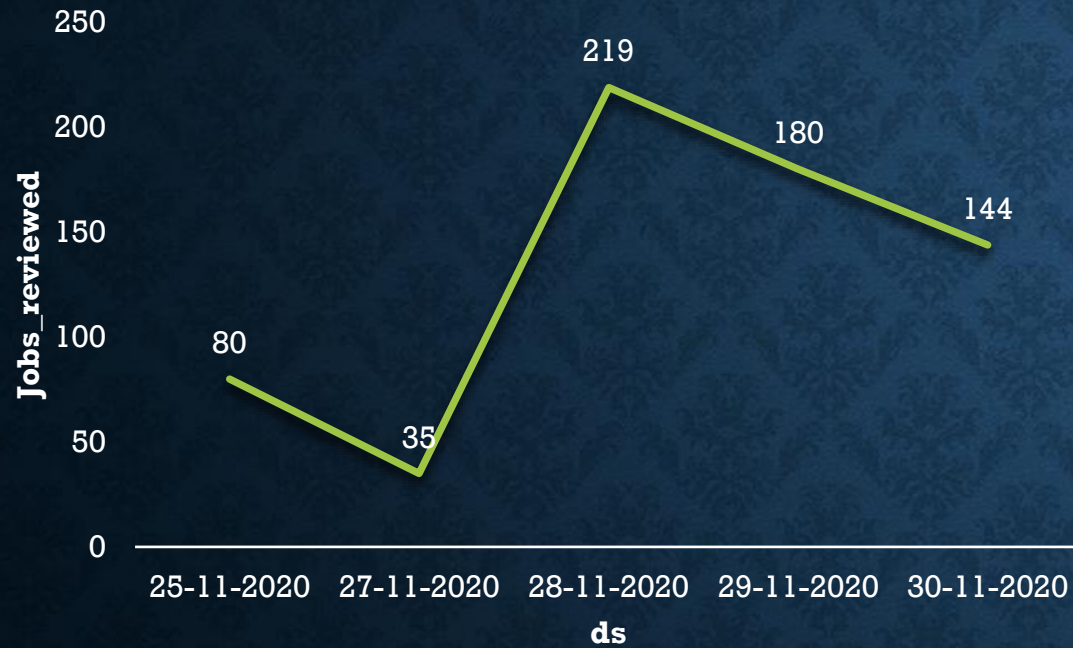
2.Throughput: it is the no. Of events happening per second.

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?

```
select ds,round(avg(s.throughput) over(order by s.ds asc rows between 6 preceding and current row),2) as rolling_7_average
from
(select ds, ceil(count(job_id)*3600/sum(time_spent)) as throughput
from oa.job_data
where event in('transfer','decision')
and ds between '1-11-2020' and '30-11-2020'
group by 1)as s;
```



DAILY METRIC VS 7 DAY AVERAGE

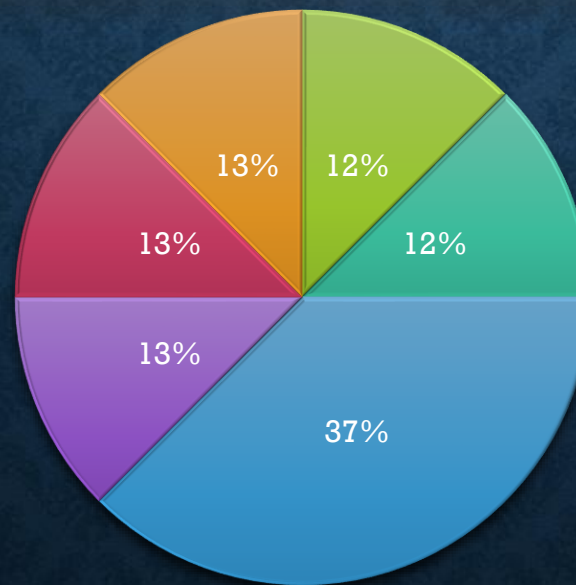


We should prefer daily metric as metric spike for job review is more in daily metric than 7 day average and overall job reviewed in daily metric is more .

3. Percentage share of each language: share of each language for different contents.

Task: calculate the percentage share of each language in the last 30 days?

```
select s1.language , 100* s1.count/s1.sum as percentage_share
from
(select s.language , s.count, sum(s.count) over() as sum
from
(select language, count(language) as count FROM oa.job_data
where ds between '1-11-2020' and '30-11-2020'
group by 1) as s
) as s1
```



English Arabic Persian Hindi French Italian

4.Duplicate rows: rows that have the same value present in them.

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

```
select job_id,actor_id,event,language,time_spent,org,ds from  
(select *, row_number() over(partition by job_id,language)as rn  
from oa.job_data)as s  
where rn >1
```

job_id	actor_id	event	Language	time_spent	org	ds
23	1005	transfer	Persian	22	D	28-11-2020
23	1004	skip	Persian	56	A	26-11-2020

CASE STUDY - 2

Operation-2 (Table-1 (Users))

user_id	A unique ID per user. Can be joined to user_id in either of the other tables.
created_at	The time the user was created (first signed up)
state	The state of the user (active or pending)
activated_at	The time the user was activated, if they are active
company_id	The ID of the user's company
language	The chosen language of the user

Operation-2 (Table-3(email_events))

user_id	The ID of the user to whom the event relates. Can be joined to user_id in either of the other tables.
occurred_at	The time the event occurred.
action	The name of the event that occurred. "sent_weekly_digest" means that the user was delivered a digest email showing relevant conversations from the previous day. "email_open" means that the user opened the email. "email_clickthrough" means that the user clicked a link in the email.

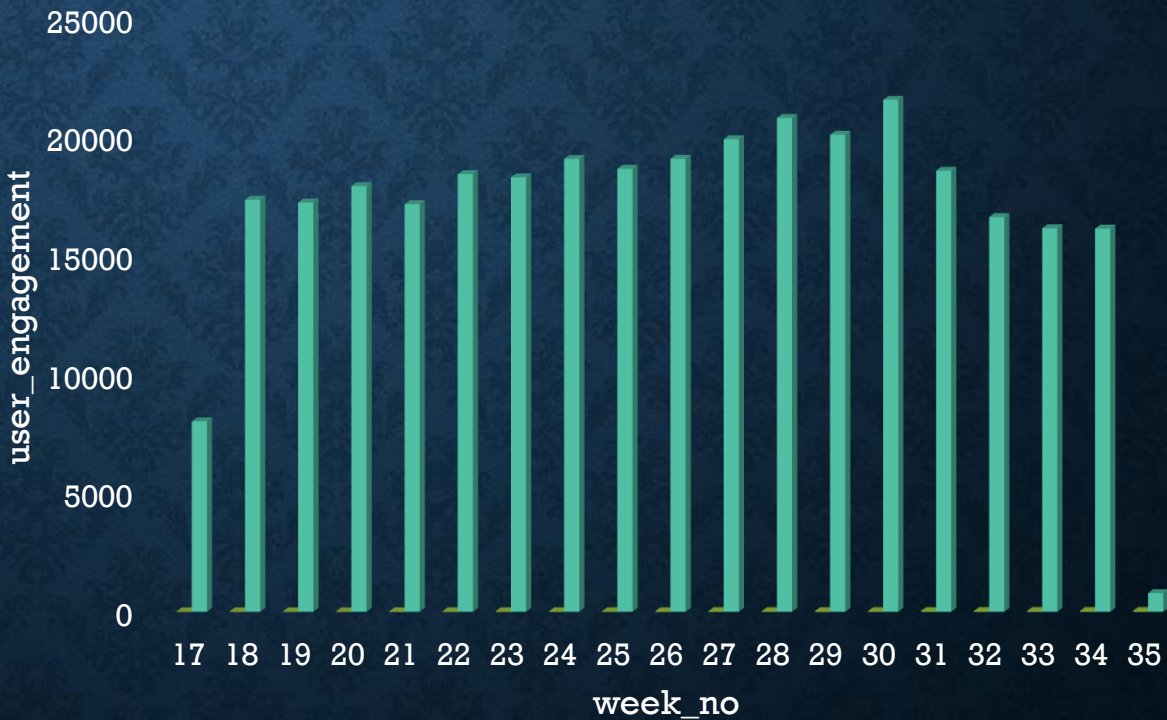
Operation-2 (Table-2(events))

user_id	The ID of the user logging the event. Can be joined to user_id in either of the other tables.
occurred_at	The time the event occurred.
event_type	The general event type. There are two values in this dataset: "signup_flow", which refers to anything occurring during the process of a user's authentication, and "engagement", which refers to general product usage after the user has signed up for the first time
event_name	The specific action the user took. Possible values include: create_user: User is added to Yammer's database during signup process enter_email: User begins the signup process by entering her email address enter_info: User enters her name and personal information during signup process complete_signup: User completes the entire signup/authentication process home_page: User loads the home page like_message: User likes another user's message login: User logs into Yammer search_autocomplete: User selects a search result from the autocomplete list search_run: User runs a search query and is taken to the search results page search_click_result_X: User clicks search result X on the results page, where X is a number from 1 through 10. send_message: User posts a message view_inbox: User views messages in her inbox
location:	The country from which the event was logged (collected through IP address).
device:	The type of device used to log the event.

1.User engagement: to measure the activeness of a user. Measuring if the user finds quality in a product/service.
Task: calculate the weekly user engagement?

```
SELECT week(occurred_at) as week_num,count(user_id) as user_engagement FROM ms.events as e
join ms.users as u
using (user_id)
where event_type='engagement' and activated_at is not null and state = 'active'
group by 1
order by 1;
```

week_no	user_engagement
17	8019
18	17341
19	17224
20	17911
21	17151
22	18413
23	18280
24	19052
25	18642
26	19061
27	19881
28	20776
29	20067
30	21533
31	18556
32	16612
33	16145
34	16127
35	784

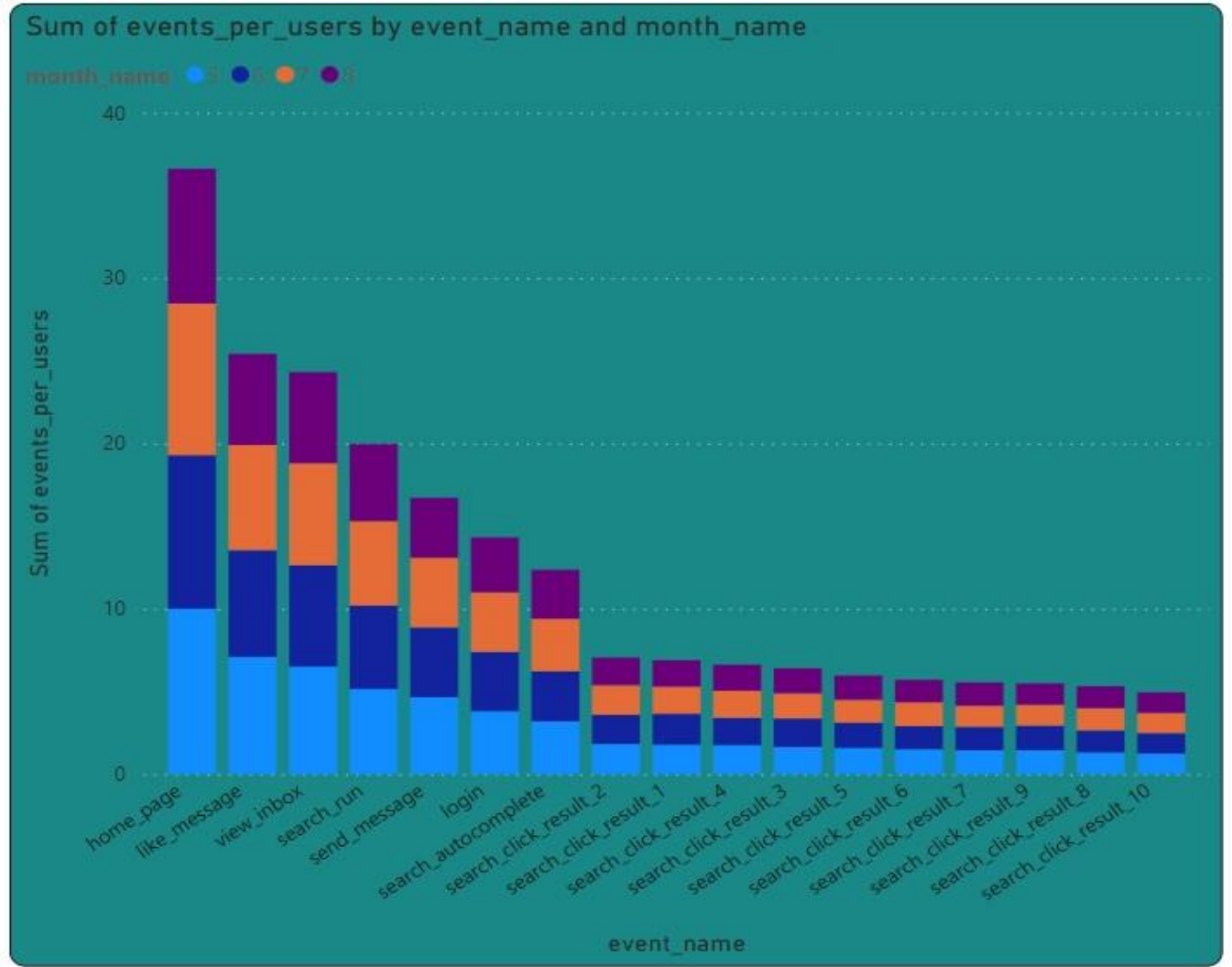


2.User growth: amount of users growing over time for a product.

Task: calculate the user growth for product?

```
SELECT
month(occurred_at) as month_num, event_name,
count(event_name) as count_events,
count(DISTINCT user_id) as count_users,
count(event_name)/count(DISTINCT user_id) as events_per_user
FROM ms.events as e
join ms.users as u
using (user_id)
where activated_at is not null and state = 'active' and event_type = 'engagement'
GROUP BY 1,2
order by 2,1;
```


OUTPUT:-



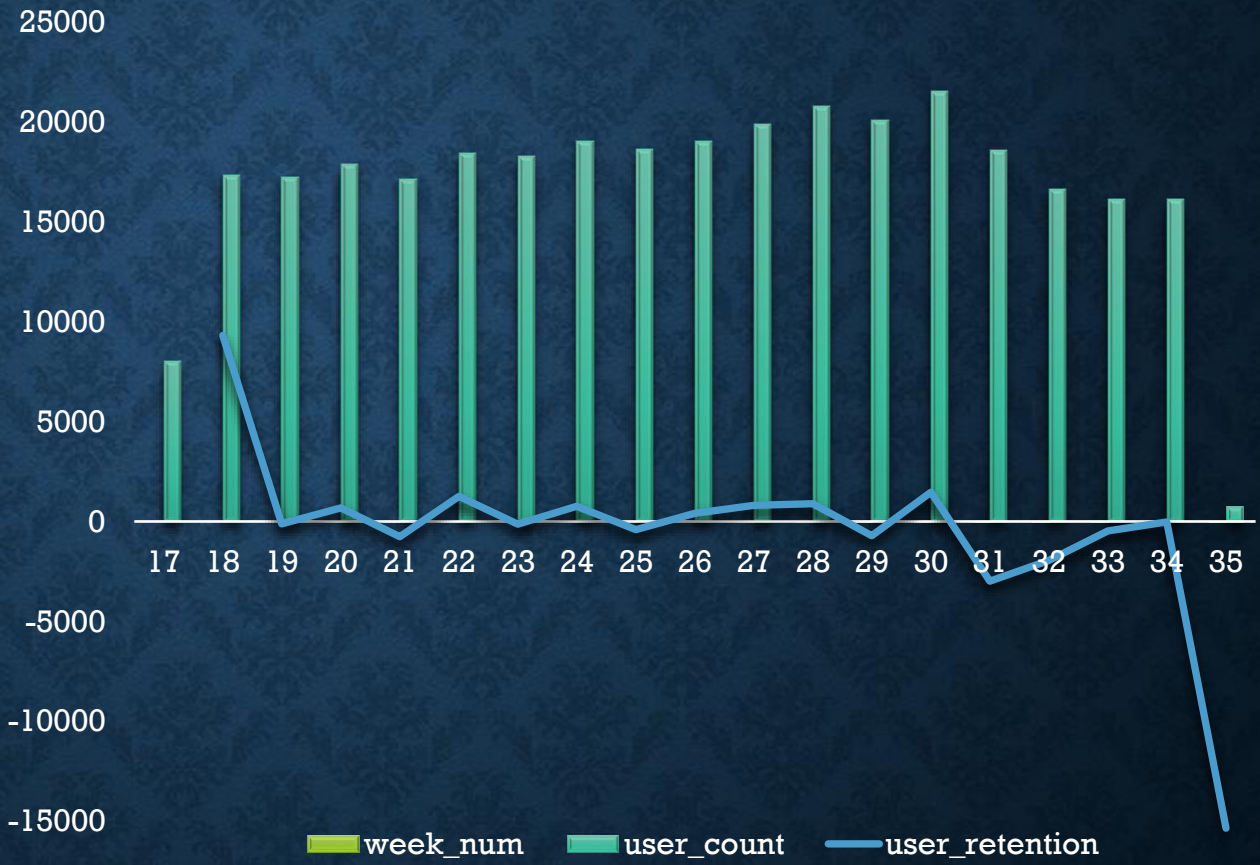
event_name	5	6	7	8	Total
home_page	10.00	9.28	9.18	8.14	36.60
like_message	7.08	6.46	6.36	5.51	25.41
login	3.80	3.58	3.60	3.32	14.31
search_autocomplete	3.18	3.04	3.17	2.96	12.35
search_click_result_1	1.78	1.87	1.64	1.59	6.88
search_click_result_10	1.24	1.24	1.23	1.24	4.94
search_click_result_2	1.82	1.76	1.80	1.66	7.04
search_click_result_3	1.63	1.73	1.53	1.51	6.39
search_click_result_4	1.75	1.65	1.63	1.58	6.61
search_click_result_5	1.58	1.53	1.39	1.46	5.95
search_click_result_6	1.50	1.39	1.44	1.36	5.70
search_click_result_7	1.44	1.42	1.27	1.40	5.54
search_click_result_8	1.30	1.34	1.34	1.32	5.31
search_click_result_9	1.44	1.48	1.27	1.28	5.48
search_run	5.15	5.04	5.10	4.66	19.95
send_message	4.65	4.22	4.21	3.63	16.71
view_inbox	6.51	6.12	6.17	5.51	24.31
Total	55.86	53.16	52.34	48.12	209.48

3. Weekly retention: users getting retained weekly after signing-up for a product.
Task: calculate the weekly retention of users-sign up cohort?

```
with cte as (  
  SELECT  
    week(occurred_at) as week_num,  
    count(user_id) as user_count  
  FROM ms.events as e  
  join ms.users as u  
  using (user_id)  
  where activated_at is not null and state = 'active' and event_type = 'engagement'  
  GROUP BY 1  
)  
SELECT  
  *,  
  (user_count - LAG(user_count) OVER (ORDER BY week_num ASC)) as user_retention  
FROM cte
```


OUTPUT:-

week_num	user_count	user_retention
17	8019	
18	17341	9322
19	17224	-117
20	17911	687
21	17151	-760
22	18413	1262
23	18280	-133
24	19052	772
25	18642	-410
26	19061	419
27	19881	820
28	20776	895
29	20067	-709
30	21533	1466
31	18556	-2977
32	16612	-1944
33	16145	-467
34	16127	-18
35	784	-15343



4. Weekly engagement: to measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.

Task: calculate the weekly engagement per device?

```
select s.device,s.week_num , count(*) as weekly_engagement_per_device
from
(
    SELECT u.user_id,u.state,u.activated_at,e.event_type,e.device,week(e.occurred_at)as week_num
    FROM ms.users as u
    join
    ms.events as e
    using (user_id)
    order by 1 asc
) as s
where s.state = 'active' and activated_at is not null
and s.event_type = 'engagement'
group by 1,2
order by 1,2 asc;
```


OUTPUT:-

device	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	Total
macbook pro	1516	3301	3159	3097	3044	3046	3123	3028	2932	3309	3548	3461	3155	3578	3608	3320	3182	3141	122	56670
lenovo thinkpad	793	1732	2143	2203	1893	1803	1882	1806	2096	2214	2233	2564	2438	2584	2114	1898	2156	1908	123	36583
macbook air	490	1604	1331	1443	1269	1714	1431	1716	1269	1489	1652	1671	1691	1731	1574	1265	1622	1456	64	26482
iphone 5	706	1328	1190	1302	1567	1394	1748	1475	1645	1602	1867	1654	1593	1535	1491	1301	1203	966	9	25576
dell inspiron notebook	503	953	1193	975	955	1011	1034	1082	1209	1077	992	1183	1179	1488	1242	1229	1011	1088	66	19470
samsung galaxy s4	449	1130	1024	1001	960	1020	1094	940	1102	1114	1154	1260	1462	1203	1068	812	729	885	29	18436
nexus 5	382	938	944	1278	1020	1231	997	1151	935	912	1015	946	793	983	684	678	641	770	34	16332
iphone 5s	473	778	964	1024	804	920	859	959	963	1026	952	1039	932	1164	663	775	712	715	22	15744
dell inspiron desktop	187	683	444	504	568	737	494	713	644	672	541	744	560	685	465	519	408	465	4	10037
iphone 4s	217	448	546	608	522	477	459	609	442	503	720	783	605	740	631	318	349	472	57	9506
asus chromebook	251	523	268	461	550	631	671	471	432	632	510	511	529	565	636	611	581	557	38	9428
ipad air	330	520	595	611	428	591	404	619	640	580	482	547	620	656	573	463	376	306		9341
acer aspire notebook	206	363	406	483	462	431	449	509	605	328	582	541	592	646	554	580	498	561	28	8824
hp pavilion desktop	132	373	376	276	475	371	655	721	588	512	589	622	742	466	601	549	379	351	10	8788
nexus 7	177	252	334	372	211	433	303	418	557	428	366	342	392	629	407	235	255	328	17	6456
nokia lumia 635	128	341	215	151	190	308	273	489	446	456	325	405	461	333	298	357	215	151	7	5549
ipad mini	205	309	381	264	280	342	258	313	234	464	344	364	364	373	240	257	260	252	21	5525
acer aspire desktop	67	295	242	226	328	255	240	289	263	313	296	303	212	403	383	354	363	281	7	5120
nexus 10	145	370	232	217	258	311	533	369	258	261	320	261	272	282	274	318	163	208	15	5067
mac mini	59	159	255	272	281	276	170	298	230	140	169	340	342	316	232	120	331	395	25	4410
htc one	190	174	270	365	260	323	234	136	285	225	225	318	227	306	117	146	138	271	18	4228
kindle fire	57	265	225	242	355	231	352	239	209	254	268	342	293	234	136	67	120	119	32	4040
windows surface	87	107	163	194	183	189	155	209	234	177	345	304	260	170	216	86	130	160	30	3395
samsung galaxy note	116	139	117	160	196	252	183	245	134	97	131	118	141	156	112	116	127	96	6	2642
amazon fire phone	83	177	141	104	29	46	181	147	131	137	109	51	92	170	157	161	114	106		2136
samsung galaxy tablet	70	79	66	78	63	70	98	101	159	139	146	102	120	137	80	77	82	119		1786
Total	8019	17341	17224	17911	17151	18413	18280	19052	18642	19061	19881	20776	20067	21533	18556	16612	16145	16127	784	321575

Since the output was very large in number and was not insightful I classified devices into PC, smartPhones and Tablets for better insights.

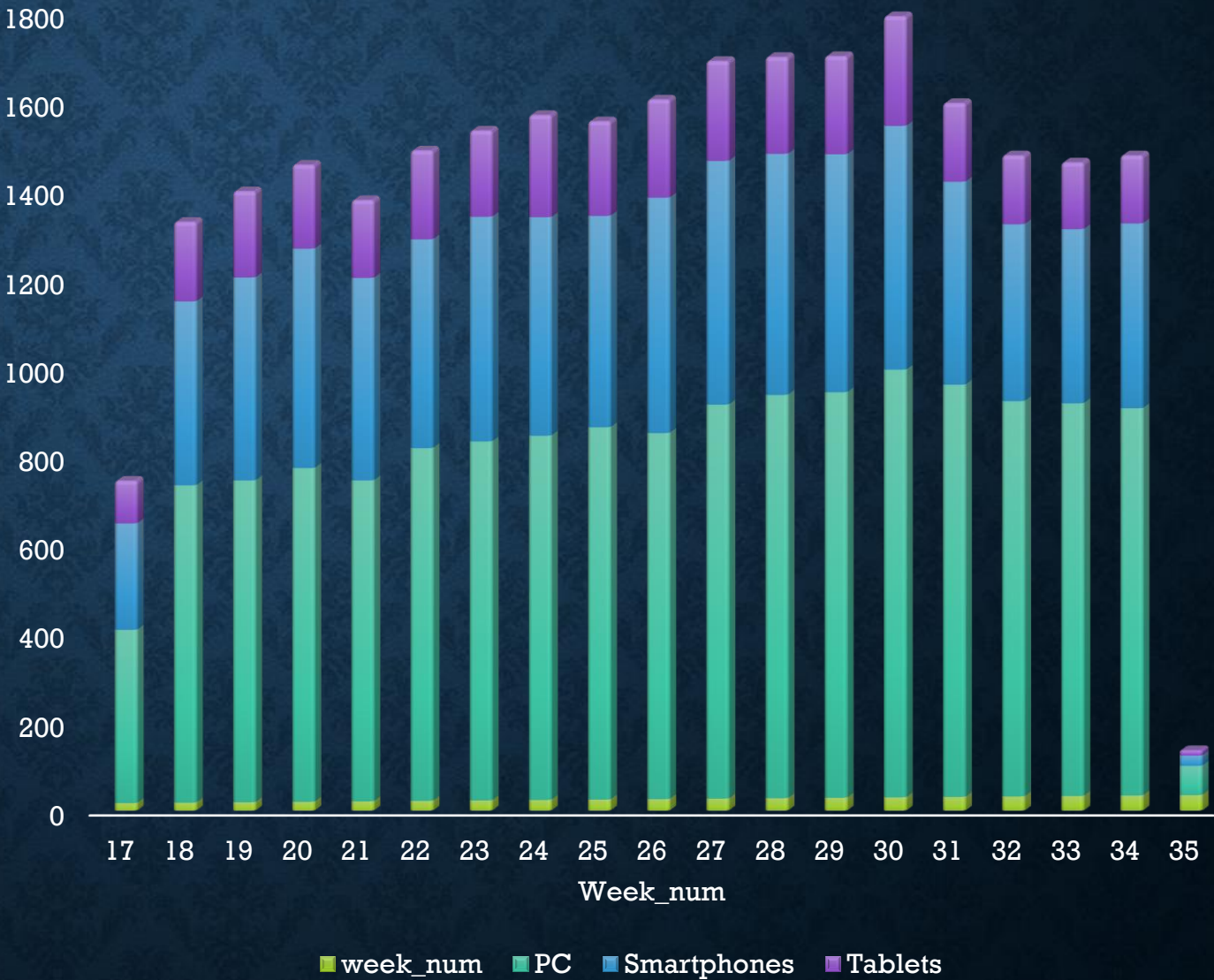

```

select week_num ,COUNT(DISTINCT CASE WHEN device IN ('dell inspiron notebook','lenovo thinkpad','macbook air',
'macbook pro','acer aspire notebook','asus chromebook','acer aspire desktop','mac mini',
'hp pavilion desktop','dell inspiron desktop') THEN user_id ELSE NULL END) AS PC,
count(DISTINCT CASE WHEN device IN('iphone 5','iphone 4s','iphone 5s','nexus 5','samsung galaxy s4','htc one',
'nokia lumia 635','samsung galaxy note','amazon fire phone') THEN user_id ELSE NULL END) AS smartphone,
COUNT(DISTINCT CASE WHEN device IN ('ipad air', 'nexus 7','ipad mini', 'nexus 10','kindle fire',
'windows surface','samsung galaxy tablet') THEN user_id ELSE NULL END) AS tablets
from
(
    SELECT u.user_id,u.state,u.activated_at,e.event_type,e.device,week(e.occurred_at)as week_num
    FROM ms.users as u
    join
    ms.events as e
    using (user_id)
    order by 1 asc
) as s
where s.state = 'active' and activated_at is not null
and s.event_type = 'engagement'
group by 1
order by 1 asc;

```


OUTPUT:-

week_num	PC	Smartphones	Tablets
17	391	240	96
18	716	415	179
19	726	458	194
20	753	495	189
21	724	457	175
22	796	471	200
23	810	507	194
24	822	493	230
25	840	477	213
26	826	531	221
27	889	550	225
28	910	544	218
29	915	537	221
30	965	550	247
31	930	458	177
32	892	399	155
33	886	393	150
34	874	417	153
35	66	23	12



Email actions which comes under engagement event_type

```
SELECT distinct em.action ,e.event_type
FROM ms.email_events as em
join ms.events as e                                -- to find which email action comes under engagement
using (user_id)
join ms.users as u
using (user_id)
where event_type='engagement' and state = 'active' and activated_at is not null
```

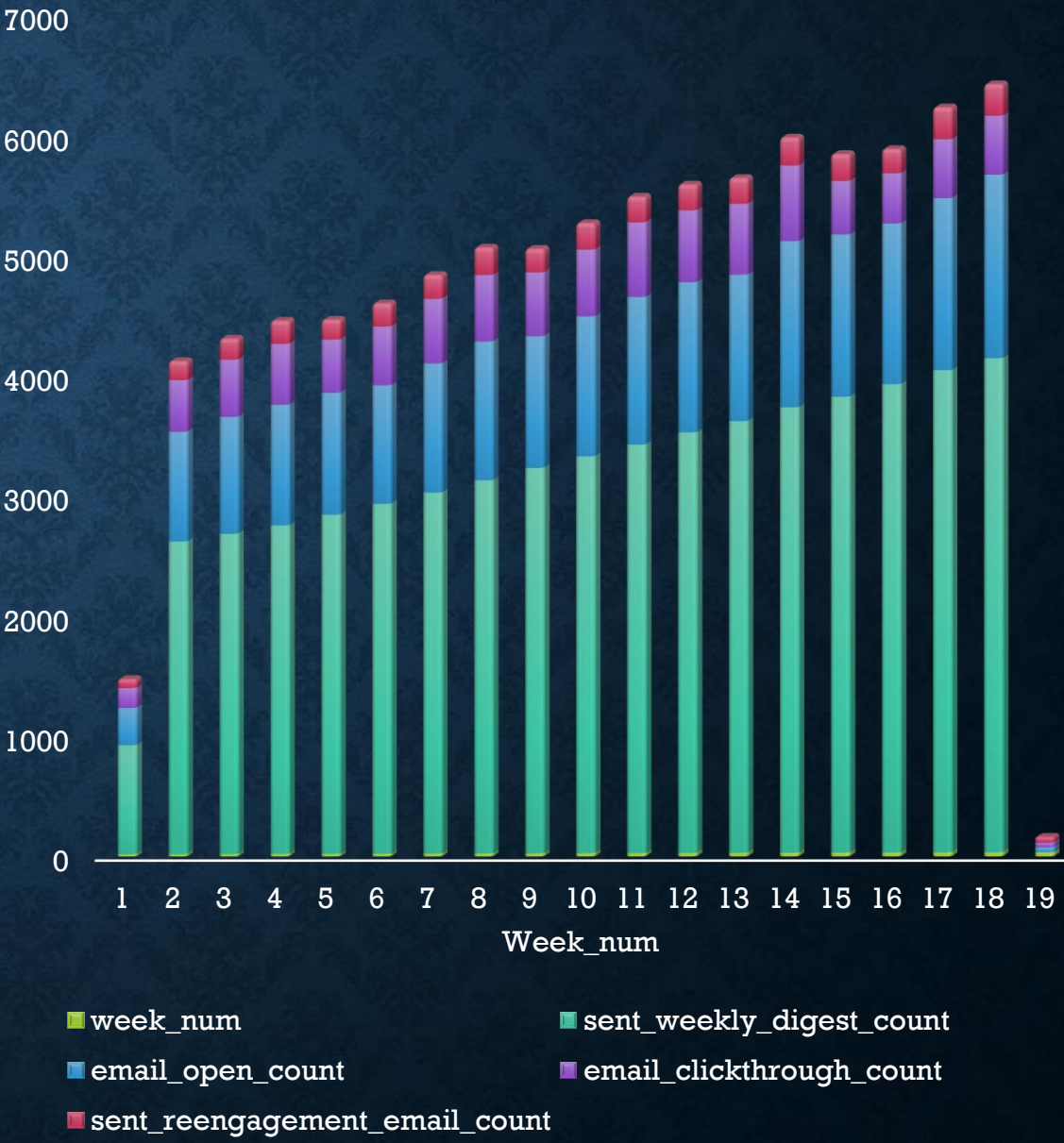
action	event_type
sent_weekly_digest	engagement
email_open	engagement
email_clickthrough	engagement
sent_reengagement_email	engagement

5.Email engagement: users engaging with the email service.
Task: calculate the email engagement metrics?

```
SELECT week(em.occurred_at) as week_num ,count(case when action = 'sent_weekly_digest'
then em.user_id else null end ) as sent_weekly_digest_count,
count(case when action = 'email_open' then em.user_id else null end ) as email_open_count,
count(case when action = 'email_clickthrough' then em.user_id else null end ) as email_clickthrough_count,
count(case when action = 'sent_reengagement_email' then em.user_id else null end ) as sent_reengagement_email_count
FROM ms.email_events as em
group by 1
order by 1;
```

OUTPUT:-

week_num	sent_weekly_digest_count	email_open_count	email_clickthrough_count	sent_reengagement_email_count
17	908	310	166	73
18	2602	912	430	157
19	2665	972	477	173
20	2733	1004	507	191
21	2822	1014	443	164
22	2911	987	488	192
23	3003	1075	538	197
24	3105	1155	554	226
25	3207	1096	530	196
26	3302	1165	556	219
27	3399	1228	621	213
28	3499	1250	599	213
29	3592	1219	590	213
30	3706	1383	630	231
31	3793	1351	445	222
32	3897	1337	418	200
33	4012	1432	490	264
34	4111	1528	490	261
35	0	41	38	48



THANK YOU