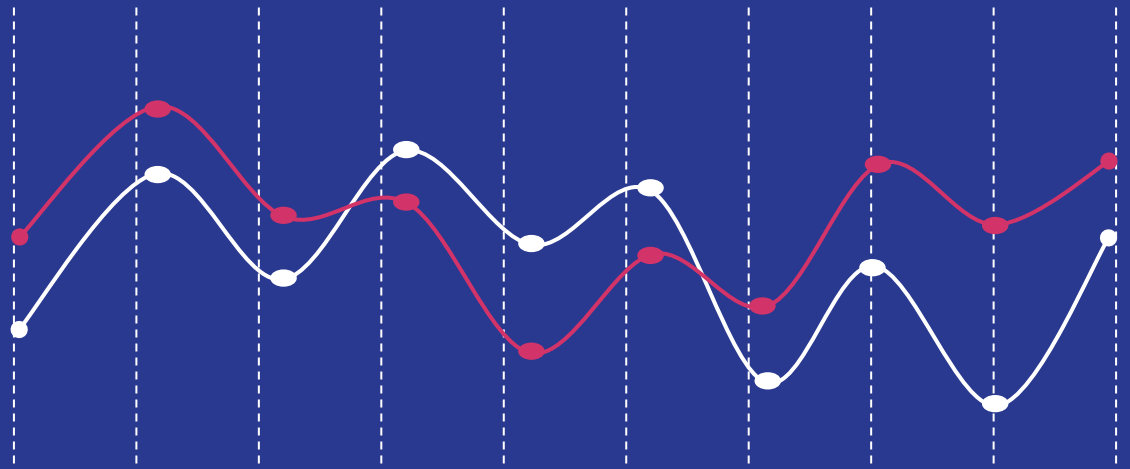
Operation Analytics and Investigating Metric Spike



Project Description

- The given project consists of 2 case studies:-
- First is regarding Operation Analytics where job data is provided and number of jobs reviewed , 7day rolling average of throughput, percentage share of language used and duplicates are found out.
- Second is Investigating Metric Spike where user engagement, user growth, weekly retention, weekly engagement and email engagement is determined.
- The following information is found with the help of SQL queries.

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.

Tech stack used

- MySQL was used to run the queries.
- The language was selected because of comfort and experience in the same.
- MS Excel was used in the second case study for better visualisation.
- As I am currently learning this tool, it was utilised to get more hands on experience.

Case-1: operation analytics

Insights

Insights-Number of jobs reviewed

```
select
avg(t) as 'avg jobs reviewed per day per hour',
avg(p) as 'avg jobs reviewed per day per second'
From
(select ds,((count(job_id)*3600)/sum(time_spent))
as t,
((count(job_id))/sum(time_spent)) as p
from job_data
where month(ds)=11
group by ds) a;
```

avg jobs reviewed per day per hour	avg jobs reviewed per day per second
126.1804833	0.03505

Insight-throughput and 7-day rolling average of throughput

```
Selectds,
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;
```

7 day rolling average is better because it can offset the throughput fluctuations of one day and create a more accurate picture

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

Insight-percentage share of language used in last 30 days

```
with a as
(select max(ds)  as m from job_data)
select distinctlanguage,
(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 aWhere
datediff(m,date(ds)) between 0 and 30)a1;
```

language	percentage
Italian	12.5
Persian	37.5
French	12.5
Hindi	12.5
Arabic	12.5
English	12.5

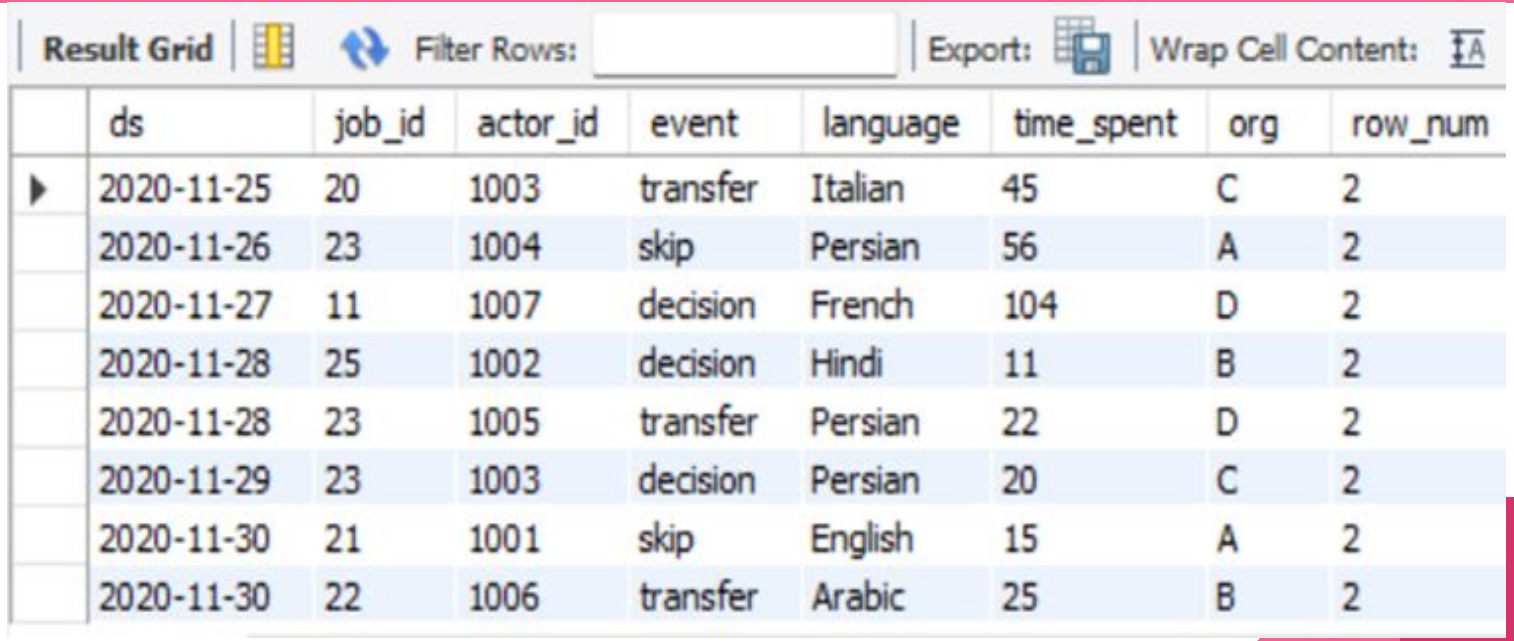
Insight-finding duplicates

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

When no duplicate data

	ds	job_id	actor_id	event	language	time_spent	org	row_num

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

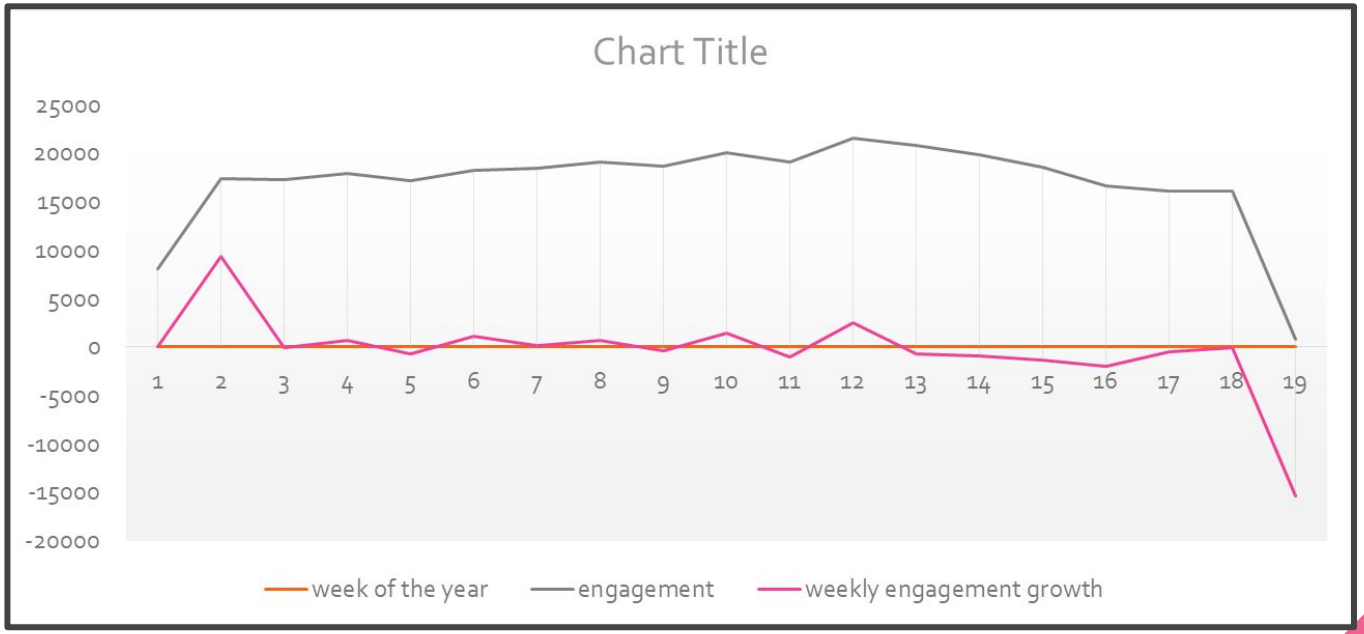


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

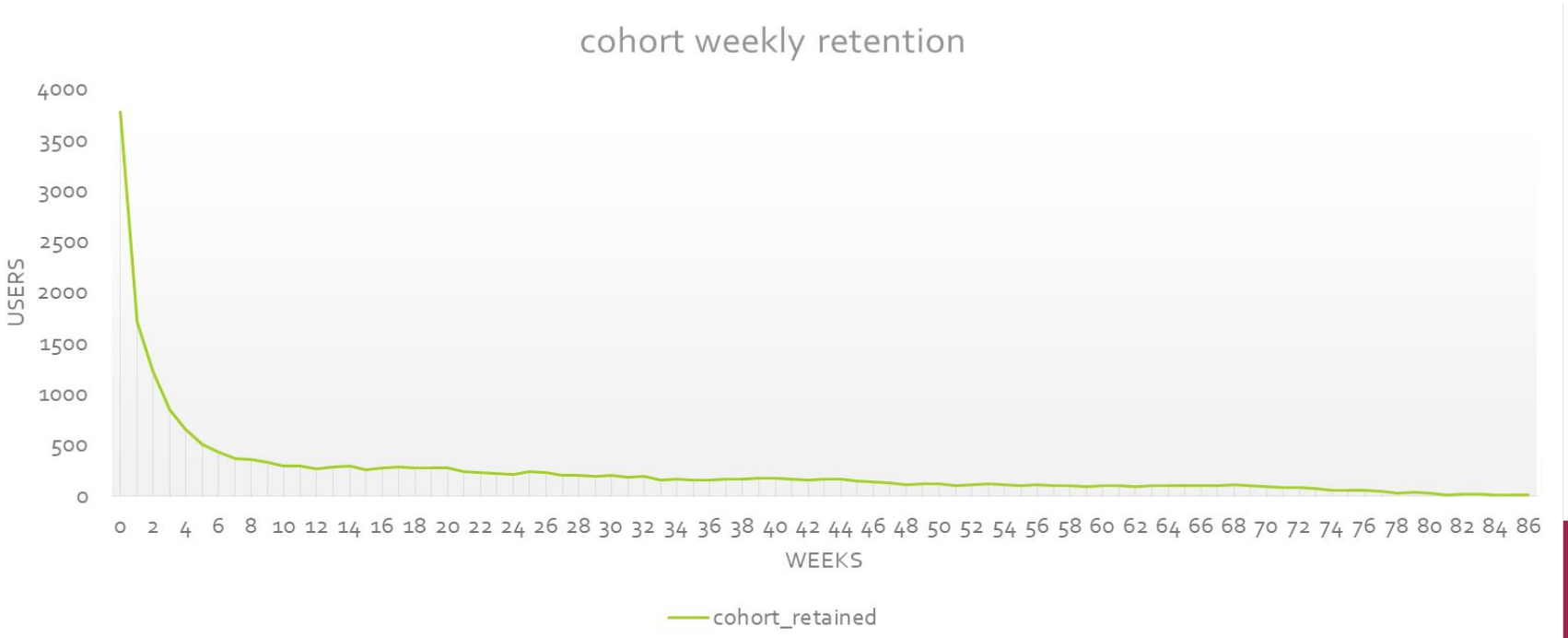
Case-2: Investigating Metric Spike Insights

Insight weekly user engagement



Overall Reduction in the engagement is seen (*note:-the data of the 35th should not be considered as it is only for the first day of the week) (query on next slide)**

Insight-weekly retention cohort analysis



Major drop in the first 10 weeks at the end of 85 weeks only 2 users remain

Insight-weekly retention cohort analysis

```
Selectweek_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_retainedFrom
(select user_id, activated_at
 from users where state='active'group by 1) ainner join
(select user_id,occurred_at from events )b
 on a.user_id=b.user_id
group by 1) c;
```

Insight-weekly engagement per device

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

Samsung galaxy table is used least

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

Insight-weekly engagement per device

```
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 eventswhere event_name='login’
group by 1,2
order by 1) a
group by 1;
```

Insight-e-mail engagement metric

Overall increase in the engagement metric seen

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

Insight-e-mail engagement metric

```
Selectweek,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

- Really engaging project, difficulty of the project makes it more fulfilling to execute.
- Learnt a lot of new things like rolling average, cohort retention analysis.
- Tired to insert excel charts wherever I could, hopefully would be able to use excel more efficiently next time.
- Became better in using windows function.