OPERATION & METRIC ANALYSIS

Created by : Manthan Nimbalkar

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. 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.

APPROACH

- In order to execute the project, SQL server was used. SQL queries were used to create a database using the raw data provided. Once the database was created, various sorting and data extracting queries were used to get the data insights required
- For 1st case study Insights are get with SQL Server
- In 2nd case study there was big dataset so python is used to clean the data then imported to SQL server and gain insights from it

TECH STACK USED

- For case study 1 only used SQL Server 2014. There were no duplicate and unknown values found . I also executed some queries to find some insights
- For case study 2 first cleaned data with the help of python language in jupyter notebook then used SQL Server 2014. There were no duplicate and unknown values found . I also executed some queries to find some insights

INSIGHTS

(FROM NEXT PAGE)

CASE STUDY 1

JOB DATA

DATASET USED

- Table Name : job_data
- Columns :
  - **job_id:** unique identifier of jobs
  - **actor_id:** unique identifier of actor
  - **event:** decision/skip/transfer
  - **language:** language of the content
  - **time_spent:** time spent to review the job in seconds
  - **org:** organization of the actor
  - **ds:** date in the yyyy/mm/dd format. It is stored in the form of text and we use presto to run. no need for date function

A. NUMBER OF JOBS REVIEWED: AMOUNT OF JOBS REVIEWED OVER TIME. CALCULATE THE NUMBER OF JOBS REVIEWED PER HOUR PER DAY FOR NOVEMBER 2020?

SQL Query :

```
# Q 1 Number of jobs reviewed
SELECT ROUND(COUNT(DISTINCT job_id) / CAST(30*24 AS decimal(6,3)), 4) AS jobs_per_day
FROM job_data
WHERE ds >= '2020-11-01' AND ds <= '2020-11-30';
```

Result:

	jobs_per_d...
1	0.0306000

B. THROUGHPUT: IT IS THE NO. OF EVENTS HAPPENING PER SECOND.

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?

SQL Query :

```sql
# Q 2 Throughput
SELECT ds as date, total_events, AVG(total_events) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS rolling_avg_7_day
FROM (
    SELECT ds, COUNT(DISTINCT event) AS total_events
    FROM job_data
    WHERE ds >= '2020-11-01' AND ds <= '2020-12-31'
    GROUP BY ds
) sub;
```

Result:

	date	total_events	rolling_avg_7_day
1	2020-11-01	1	1
2	2020-11-02	1	1
3	2020-11-03	1	1
4	2020-11-04	1	1
5	2020-11-05	1	1
6	2020-11-06	1	1
7	2020-11-07	1	1
8	2020-11-08	1	1
9	2020-11-09	1	1
10	2020-11-10	1	1
11	2020-11-11	1	1
12	2020-11-12	1	1
13	2020-11-13	1	1
14	2020-11-14	1	1
15	2020-11-15	1	1
16	2020-11-16	1	1
17	2020-11-17	1	1
18	2020-11-18	1	1
19	2020-11-19	1	1
20	2020-11-20	1	1
21	2020-11-21	1	1
22	2020-11-22	1	1
23	2020-11-23	1	1
24	2020-11-24	1	1
25	2020-12-01	2	1
26	2020-12-02	1	1
27	2020-12-03	2	1
28	2020-12-04	1	1
29	2020-12-05	1	1
30	2020-12-06	1	1
31	2020-12-07	1	1
32	2020-12-08	1	1
33	2020-12-09	1	1
34	2020-12-10	1	1
35	2020-12-11	1	1
36	2020-12-12	1	1
37	2020-12-13	1	1
38	2020-12-14	1	1
39	2020-12-15	2	1
40	2020-12-16	2	1
41	2020-12-17	1	1
42	2020-12-18	1	1
43	2020-12-19	1	1
44	2020-12-20	1	1
45	2020-12-21	1	1
46	2020-12-22	1	1
47	2020-12-23	1	1
48	2020-12-24	1	1
49	2020-12-25	3	1
50	2020-12-26	3	1
51	2020-12-27	1	1
52	2020-12-28	1	1
53	2020-12-29	1	1
54	2020-12-30	1	1
55	2020-12-31	3	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?

SQL Query :

```sql
# Q 3 Percentage share of each language
select language, count(language) as total,
count(*)*100/sum(count(*)) over() as percentage
from job_data
where ds >= '2020-12-01' and ds <= '2020-12-31'
group by language
order by percentage desc;
```

Result:

	language	total	percentage
1	German	10	18
2	French	8	14
3	Marathi	8	14
4	Italian	7	12
5	Arabic	6	11
6	English	6	11
7	Hindi	5	9
8	Persian	4	7

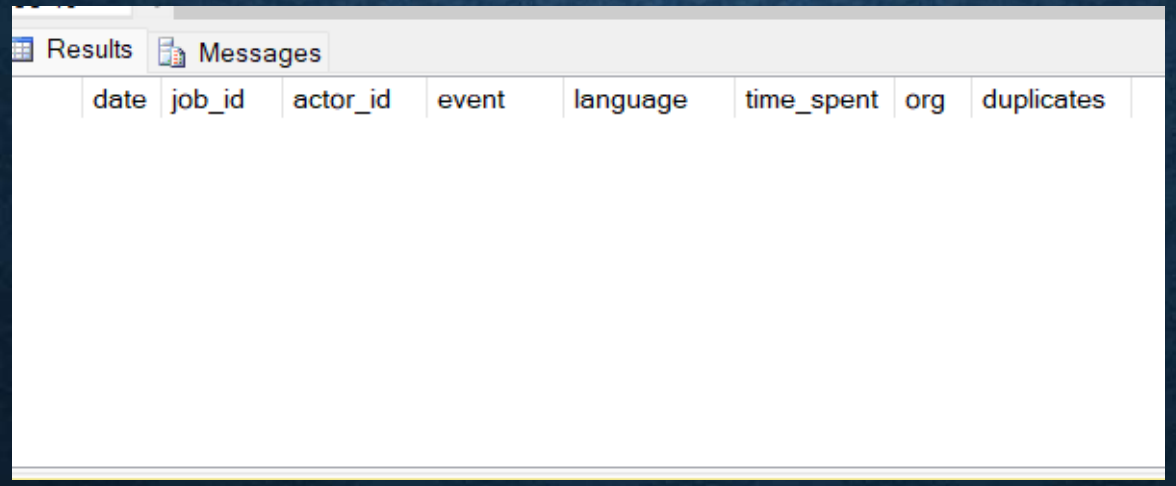
D. DUPLICATE ROWS: ROWS THAT HAVE THE SAME VALUE PRESENT IN THEM.

LET'S SAY YOU SEE SOME DUPLICATE ROWS IN THE DATA. HOW WILL YOU DISPLAY DUPLICATES FROM THE TABLE?

SQL Query :

```
# Q 4 Duplicate rows
select ds as date, job_id, actor_id, event, language, time_spent, org, count(*) as duplicates
from job_data
group by ds, job_id, actor_id, event, language, time_spent, org
having count(*) > 1;
```

Result :



CASE STUDY 2

INVESTIGATING MATRIC SPIKE

DATASET USED

- **Table-1:** users
  This table includes one row per user, with descriptive information about that user's account.

- **Table-2:** events
  This table includes one row per event, where an event is an action that a user has taken. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.

- **Table-3:** email_events
  This table contains events specific to the sending of emails. It is similar in structure to the events table above.

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?

SQL Query :

```sql
SELECT DATEPART(week, occurred_at) as week_no, COUNT(DISTINCT user_id) as no_users
FROM events
WHERE event_type = 'engagement'
GROUP BY DATEPART(week, occurred_at)
ORDER BY no_users desc;
```

Result :

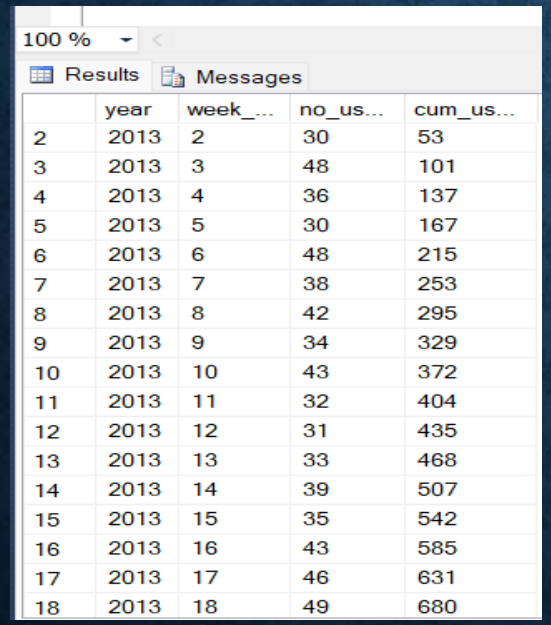
	week_no	no_users
1	31	1467
2	30	1376
3	28	1372
4	29	1365
5	27	1302
6	32	1299
7	25	1275
8	26	1264
9	24	1232
10	34	1225
11	33	1225
12	35	1204
13	23	1186
14	21	1154
15	22	1121
16	20	1113
17	19	1068
18	18	663
19	36	104

B. USER GROWTH: AMOUNT OF USERS GROWING OVER TIME FOR A PRODUCT. YOUR TASK: CALCULATE THE USER GROWTH FOR PRODUCT?

SQL Query :

```sql
SELECT
    DATEPART(year, created_at) AS year,
    DATEPART(week, created_at) AS week_no,
    COUNT(DISTINCT user_id) AS no_users,
    SUM(COUNT(DISTINCT user_id)) OVER (ORDER BY DATEPART(year, created_at), DATEPART(week, created_at)) AS cum_users
FROM users
WHERE state = 'active'
GROUP BY DATEPART(year, created_at), DATEPART(week, created_at)
ORDER BY DATEPART(year, created_at), DATEPART(week, created_at);
```

Result :



100 %

Results | Messages

	year	week_...	no_us...	cum_us...
2	2013	2	30	53
3	2013	3	48	101
4	2013	4	36	137
5	2013	5	30	167
6	2013	6	48	215
7	2013	7	38	253
8	2013	8	42	295
9	2013	9	34	329
10	2013	10	43	372
11	2013	11	32	404
12	2013	12	31	435
13	2013	13	33	468
14	2013	14	39	507
15	2013	15	35	542
16	2013	16	43	585
17	2013	17	46	631
18	2013	18	49	680

Total Results 89 Rows

Week no 34 of year 2014 has seen highest no of users whereas Week no 36 of year 2014 has lowest no users

C. WEEKLY RETENTION: USERS GETTING RETAINED WEEKLY AFTER SIGNING-UP FOR A PRODUCT.
YOUR TASK: CALCULATE THE WEEKLY RETENTION OF USERS-SIGN UP COHORT?

SQL Query :

```
SELECT DATEPART(week, occurred_at) as week_no,
       COUNT(CASE WHEN event_type = 'engagement' THEN user_id END) as engagement,
       COUNT(CASE WHEN event_type = 'signup_flow' THEN user_id END) as signup
FROM events
GROUP BY DATEPART(week, occurred_at)
ORDER BY signup desc;
```

Result :

	week_no	engagement	sign...
1	35	16127	1339
2	34	16145	1300
3	33	16612	1260
4	31	21533	1242
5	30	20067	1166
6	25	19052	1158
7	28	19881	1140
8	29	20776	1132
9	26	18642	1075
10	27	19061	1065
11	24	18280	1065
12	23	18413	1042
13	32	18556	1029
14	22	17151	961
15	21	17911	955
16	20	17224	954
17	19	17341	901
18	18	8019	385
19	36	784	88

SQL Query :

```sql
WITH cte1 AS (
    SELECT DISTINCT user_id, DATEPART(week, occurred_at) AS signup_week
    FROM events
    WHERE event_type = 'signup_flow' AND event_name = 'complete_signup' AND DATEPART(week, occurred_at) between 18 and 36
),
cte2 AS (
    SELECT DISTINCT user_id, DATEPART(week, occurred_at) AS engagement_week
    FROM events
    WHERE event_type = 'engagement'
)
SELECT COUNT(user_id) AS total_engaged,
       SUM(CASE WHEN retention_week > 0 THEN 1 ELSE 0 END) AS retained_users
FROM (
    SELECT a.user_id, a.signup_week, b.engagement_week, b.engagement_week - a.signup_week AS retention_week
    FROM cte1 a
    LEFT JOIN cte2 b ON a.user_id = b.user_id
) sub;
```

Result :

Results | Messages

	total_engaged	retained_users
1	10534	6854

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?

SQL Query :

```
SELECT DATEPART(month, occurred_at) AS month_no, DATEPART(week, occurred_at) AS week_no, device, COUNT(user_id) as total_users
FROM events
GROUP BY DATEPART(month, occurred_at), DATEPART(week, occurred_at), device
ORDER BY total_users desc;
```

Result:

	month_...	week_...	device	total_users
1	8	32	macbook pro	3818
2	7	28	macbook pro	3767
3	7	29	macbook pro	3651
4	8	33	macbook pro	3590
5	8	34	macbook pro	3476
6	5	19	macbook pro	3469
7	8	35	macbook pro	3360
8	7	30	macbook pro	3340
9	5	20	macbook pro	3315
10	6	24	macbook pro	3286
11	5	21	macbook pro	3268
12	6	25	macbook pro	3213
13	6	23	macbook pro	3194
14	5	22	macbook pro	3173
15	6	26	macbook pro	3111

E. EMAIL ENGAGEMENT: USERS ENGAGING WITH THE EMAIL SERVICE. YOUR TASK: CALCULATE THE EMAIL ENGAGEMENT METRICS?

SQL Query :

```
SELECT
    action,
    DATEPART(month, occurred_at) as month_no,
    COUNT(action) as emails
FROM email_events
GROUP BY action, DATEPART(month, occurred_at)
ORDER BY emails desc;
```

Result:

	action	month_no	emails
1	sent_weekly_digest	8	16480
2	sent_weekly_digest	7	15902
3	sent_weekly_digest	6	13155
4	sent_weekly_digest	5	11730
5	email_open	8	5978
6	email_open	7	5611
7	email_open	6	4658
8	email_open	5	4212
9	email_clickthrough	7	2721
10	email_clickthrough	6	2274
11	email_clickthrough	5	2023
12	email_clickthrough	8	1992
13	sent_reengagement_email	8	1073
14	sent_reengagement_email	7	933
15	sent_reengagement_email	6	889
16	sent_reengagement_email	5	758

RESULTS MODULE-1

1. During the month of November approx. 3 jobs were reviewed each hour of day. Which appears to be a small numbers.
2. Top 3 languages are German, French and Marathi
3. There are no duplicate rows found in dataset

RESULTS MODULE-2

1. During 31st week highest no of engagement has been seen whereas during 36th week there was lowest no of engangement found.
2. The 34th week of 2014 has seen greatest no of active users while 36th week of 2014 has seen lowest no active users engaging with product/service.
3. The 35th week is a week where greatest no of users are retained.
4. Total 6854 users are retained
5. The MacBook Pro is most frequently used device by users each week.
6. During month of August highest no of weekly digest emails received.

Thank You