Operations Analytics and Investigating Metric Spike

Project Description:

The project aims is divided into two parts:-

1) To analyze the efficiency of work done and the most frequent language used while ensuring no duplicate rows
2) To analyze user engagement on a weekly basis

Approach:

1)

(a) Jobs Reviewed Over Time

To find the efficiency of jobs reviewed for each day in November 2020, a group by is done on the date column (here, ds), and a "where" condition is used to ensure it's done for each day in November only. A final division of total "job_id" by total "time_spent" that day to find jobs reviewed per hour per day.

Query:

```
select * from job_data;
#A
SELECT
    ds,
    SUM(time_spent) / 60 AS review_hour,
    COUNT(job_id) * 60 / SUM(time_spent) AS jobs_reviewed_per_hour_per_day
FROM
    job_data
WHERE
    ds LIKE '11%2020'
GROUP BY ds
ORDER BY ds;
```

Result:

ds	review_hour	jobs_reviewed_per_hour_per_day
11/25/2020	0.7500	1.3333
11/26/2020	0.9333	1.0714
11/27/2020	1.7333	0.5769
11/28/2020	0.5500	3.6364
11/29/2020	0.3333	3.0000
11/30/2020	0.6667	3.0000

(b) Throughput analysis

To calculate the 7-day rolling average of throughput (no. of events per second), a group by is done on the date column (here, ds), and a count of each event that happened in that day is divided by the total "time_spent" on that event.

Query:

```
SELECT
    ds, AVG(events_per_second) AS rolling_avg_7_day
FROM
    (SELECT
        ds, COUNT(*) / SUM(time_spent) AS events_per_second
    FROM
        job_data
    GROUP BY ds) AS subquery
GROUP BY ds
ORDER BY ds;
```

Result:

ds	rolling_avg_7_day
11/25/2020	0.02220000
11/26/2020	0.01790000
11/27/2020	0.00960000
11/28/2020	0.06060000
11/29/2020	0.05000000
11/30/2020	0.05000000

(c) Language Share Analysis

To figure out the most popular language, the count of each language is divided by the total languages spoken and to

bring the terms in percentage, the result is multiplied by 100.

Query:

```
#% share of each language in the last 30 days
SELECT
    `language`,
    COUNT(`language`) * 100 / (SELECT
            COUNT(*)
        FROM
            job_data) AS `% share`
FROM
    job_data
GROUP BY `language`;
```

Result:

	language	% share
▶	English	12.5000
	Arabic	12.5000
	Persian	37.5000
	Hindi	12.5000
	French	12.5000
	Italian	12.5000

(d) Duplicate Rows Detection

To find out the number of duplicate rows, the “having” clause is used to ensure each row has unique attributes in each column from left to right.

Query:

```
#duplicate rows
SELECT
    *
FROM
    job_data
GROUP BY
    ds, job_id, actor_id, `event`, `language`, time_spent, org
HAVING
    COUNT(*) > 1;
```

Result:

ds	job_id	actor_id	event	language	time_spent	org

2)

(a) Weekly User Engagement

To calculate weekly user engagement based on the participation of individuals in an " engagement" event, a count of distinct "user_id" and a count of total "event_name" is done. A group by is done after extracting week from the "occurred_at" column to complete the query in a well-organised manner.

Query:

```
#weekly user engagement
SELECT
    dayofweek(STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')) AS week_name,
    COUNT(distinct user_id) AS user_activity,
    COUNT(event_name) AS total_events
FROM
    `events` AS e
WHERE
    e.event_type != 'signup_flow'
GROUP BY week_name
ORDER BY week_name;
```

Result:

week_name	user_activity	total_events
1	330	2984
2	973	11202
3	1018	12551
4	1102	13232
5	1130	14124
6	1154	15539
7	434	4899

(b) User Growth Analysis

To analyze the user usage growth, a group by is done on month name and according to that count of every new_registration is taken.

Query:

```sql
#user growth for a product
SET SQL_SAFE_UPDATES = 0;
UPDATE users
SET created_at = STR_TO_DATE(created_at, '%d-%m-%Y %H:%i')
where user_id>0;
SELECT
    monthname(created_at) AS month_name,
    COUNT(*) AS new_users_joined
FROM
    users
WHERE
    state='active'
GROUP BY
    month_name
ORDER BY
    month_name;
```

Result:

month_name	new_users_joined
April	907
August	316
December	486
February	685
January	711
July	284
June	213

July	284
June	213
March	765
May	493
November	399
October	390
September	330

(c) Weekly Retention Analysis

To find out the user retention based on sign-up cohort, a group by is done on the basis of first year and the second on the basis of the quarter for a meaningful comparison. Each row's “user_id” count with the preceding one is done to find the user growth. Also, a “where” clause is used to ensure that “state” is “active” and their activation date is not null.

Query:

```
#user retention based on sign-up cohortop
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 year_, quarter_
) subquery;
```

Result:

year_	quarter_	new_user_activated	user_growth
NULL	NULL	1	NULL
2013	1	469	468
2013	2	608	139
2013	3	930	322
2013	4	1275	345
2014	1	1692	417
2014	2	1005	-687

(d) Weekly Engagement Per Device

To find the weekly user engagement per device, the weeks are extracted from the “week_name” column and a total count of all the active devices, one at a time, is done.

Query:

```
#weekly engagement per device
SELECT
    DAYOFWEEK(STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')) AS week_name,
    device,
    COUNT(*) AS weekly_engagement_per_device
FROM
    `events`
WHERE
    event_type = 'engagement'
GROUP BY week_name , device
ORDER BY week_name;
```

Result:

week_name	device	weekly_engagement_per_device
1	acer aspire desktop	16
1	acer aspire notebook	82
1	asus chromebook	99
1	dell inspiron desktop	105
1	dell inspiron notebook	141
1	hp pavilion desktop	31
1	htc one	20

16

week_name	device	weekly_engagement_per_device
1	ipad air	144
1	ipad mini	65
1	iphone 4s	244
1	iphone 5	239
1	iphone 5s	136
1	kindle fire	17
1	lenovo thinkpad	291

244

week_name	device	weekly_engagement_per_device
1	mac mini	33
1	macbook air	212
1	macbook pro	525
1	nexus 10	56
1	nexus 5	197
1	nexus 7	95
1	nokia lumia 635	17

week_name	device	weekly_engagement_per_device
1	samsumg galaxy tablet	10
1	samsung galaxy note	19
1	samsung galaxy s4	137
1	windows surface	53
2	acer aspire desktop	99
2	acer aspire notebook	285
2	amazon fire phone	177

99

week_name	device	weekly_engagement_per_device
2	asus chromebook	362
2	dell inspiron desktop	362
2	dell inspiron notebook	520
2	hp pavilion desktop	296
2	htc one	131
2	ipad air	513
2	ipad mini	134

week_name	device	weekly_engagement_per_device
2	iphone 4s	408
2	iphone 5	721
2	iphone 5s	489
2	kindle fire	101
2	lenovo thinkpad	1311
2	mac mini	169
2	macbook air	1166

week_name	device	weekly_engagement_per_device
2	macbook pro	1721
2	nexus 10	187
2	nexus 5	542
2	nexus 7	271
2	nokia lumia 635	148
2	samsumg galaxy tablet	81
2	samsung galaxy note	203

week_name	device	weekly_engagement_per_device
2	samsung galaxy s4	629
2	windows surface	176
3	acer aspire desktop	176
3	acer aspire notebook	414
3	amazon fire phone	40
3	asus chromebook	248
3	dell inspiron desktop	347

week_name	device	weekly_engagement_per_device
3	dell inspiron notebook	711
3	hp pavilion desktop	323
3	htc one	115
3	ipad air	405
3	ipad mini	209
3	iphone 4s	634
3	iphone 5	943

209

week_name	device	weekly_engagement_per_device
3	iphone 5s	721
3	kindle fire	150
3	lenovo thinkpad	1382
3	mac mini	245
3	macbook air	1143
3	macbook pro	2143
3	nexus 10	185

114

week_name	device	weekly_engagement_per_device
3	nexus 5	648
3	nexus 7	302
3	nokia lumia 635	101
3	samsumg galaxy tablet	72
3	samsung galaxy note	91
3	samsung galaxy s4	688
3	windows surface	115

week_name	device	weekly_engagement_per_device
4	lenovo thinkpad	1593
4	mac mini	243
4	macbook air	1201
4	macbook pro	2118
4	nexus 10	222
4	nexus 5	645
4	nexus 7	275

week_name	device	weekly_engagement_per_device
4	acer aspire desktop	191
4	acer aspire notebook	286
4	amazon fire phone	97
4	asus chromebook	472
4	dell inspiron desktop	438
4	dell inspiron notebook	812
4	hp pavilion desktop	265

812

week_name	device	weekly_engagement_per_device
4	htc one	102
4	ipad air	384
4	ipad mini	120
4	iphone 4s	534
4	iphone 5	1182
4	iphone 5s	677
4	kindle fire	173

week_name	device	weekly_engagement_per_device
4	nokia lumia 635	288
4	samsumg galaxy tablet	64
4	samsung galaxy note	75
4	samsung galaxy s4	597
4	windows surface	178
5	acer aspire desktop	238
5	acer aspire notebook	275

week_name	device	weekly_engagement_per_device
5	amazon fire phone	87
5	asus chromebook	425
5	dell inspiron desktop	350
5	dell inspiron notebook	854
5	hp pavilion desktop	329
5	htc one	177
5	ipad air	409

329

week_name	device	weekly_engagement_per_device
5	ipad mini	219
5	iphone 4s	538
5	iphone 5	1151
5	iphone 5s	699
5	kindle fire	215
5	lenovo thinkpad	1725
5	mac mini	191

week_name	device	weekly_engagement_per_device
5	macbook air	1380
5	macbook pro	2507
5	nexus 10	201
5	nexus 5	654
5	nexus 7	219
5	nokia lumia 635	227
5	samsumg galaxy tablet	92

week_name	device	weekly_engagement_per_device
5	samsung galaxy note	117
5	samsung galaxy s4	685
5	windows surface	160
6	acer aspire desktop	185
6	acer aspire notebook	482
6	amazon fire phone	71
6	asus chromebook	339

week_name	device	weekly_engagement_per_device
6	dell inspiron desktop	483
6	dell inspiron notebook	969
6	hp pavilion desktop	409
6	htc one	194
6	ipad air	407
6	ipad mini	270
6	iphone 4s	687

week_name	device	weekly_engagement_per_device
6	iphone 5	1332
6	iphone 5s	735
6	kindle fire	181
6	lenovo thinkpad	1611
6	mac mini	246
6	macbook air	1407
6	macbook pro	2653

week_name	device	weekly_engagement_per_device
6	nexus 10	210
6	nexus 5	777
6	nexus 7	282
6	nokia lumia 635	234
6	samsumg galaxy tablet	115
6	samsung galaxy note	157
6	samsung galaxy s4	959

week_name	device	weekly_engagement_per_device
6	windows surface	144
7	acer aspire desktop	48
7	acer aspire notebook	109
7	amazon fire phone	18
7	asus chromebook	75
7	dell inspiron desktop	164
7	dell inspiron notebook	248

week_name	device	weekly_engagement_per_device
7	hp pavilion desktop	121
7	htc one	65
7	ipad air	127
7	ipad mini	127
7	iphone 4s	167
7	iphone 5	285
7	iphone 5s	209

week_name	device	weekly_engagement_per_device
7	kindle fire	110
7	lenovo thinkpad	530
7	mac mini	55
7	macbook air	459
7	macbook pro	1009
7	nexus 10	129
7	nexus 5	330

week_name	device	weekly_engagement_per_device
7	nexus 5	330
7	nexus 7	95
7	nokia lumia 635	54
7	samsumg galaxy tablet	3
7	samsung galaxy note	72
7	samsung galaxy s4	224
7	windows surface	66

(e) Email Engagement Analysis

To find the user engagement on their emails, a total count of “action” is done based on each week, one at a time.

Query:

```sql
# email engagement metrics
SELECT
    DAYOFWEEK(STR_TO_DATE(occurred_at, '%d-%m-%Y %H:%i')) AS week_name,
    `action`,
    COUNT(`action`) AS total_users_engaged
FROM
    email_events
GROUP BY week_name , `action`
ORDER BY week_name;
```

Result:

week_name	action	total_users_engaged
2	email_clickthrough	25
2	email_open	75
2	sent_weekly_digest	239
3	email_clickthrough	10
3	email_open	28
3	sent_weekly_digest	105
4	email_clickthrough	13

week_name	action	total_users_engaged
4	email_open	45
4	sent_weekly_digest	136
5	email_clickthrough	31
5	email_open	67
5	sent_weekly_digest	216
6	email_clickthrough	12
6	email_open	36
6	email_clickthrough	12
6	email_open	36
6	sent_weekly_digest	126

Tech- Stack Used:

Here, I have used MySQL Workbench 8.0 due to its user-friendly interface and familiarity with it.

Insights:

1) The efficiency of jobs reviewed each day varies greatly.
2) The rolling throughput has slight variations each day.
3) “Persian” is the most spoken language and the others are spoken at the same level.
4) There is no duplicate row identification.

5) The lowest user activity is recorded on week name Sunday and the highest is recorded on week name Friday.
6) Each week, the activity of “Macbook_ pro” is recorded as the highest.
7) The maximum action done is “sent_weekly_digest” on Monday.

Result:

The project helped me gain a better understanding of concepts and hands-on experience of Advanced SQL.