

Project 3: Operation Analytics and Investigating Metric Spike

Description:

The Aim of this project is to analyze the company's end-to-end operations and to derive valuable insights for various teams, such as operations, support and marketing; to investigate metric spikes and to understand sudden changes in key metrics. The Objectives are,

For Case Study 1: Job Data Analysis

A) Jobs Reviewed Over Time: To Calculate the number of jobs reviewed per hour for each day in November 2020.

B) Throughput Analysis: To Calculate the 7-day rolling average of throughput (number of events per second).

C) Language Share Analysis: To Calculate the percentage share of each language in the last 30 days.

D) Duplicate Rows Detection: To Identify the duplicate rows in the data.

For Case Study 2: Investigating Metric Spike

A) Weekly User Engagement: To Measure the activeness of users on a weekly basis.

B) User Growth Analysis: To Analyze the growth of users over time for a product.

C) Weekly Retention Analysis: To Analyze the retention of users on a weekly basis after signing up for a product.

D) Weekly Engagement Per Device: To Measure the activeness of users on a weekly basis per device.

E) Email Engagement Analysis: To Analyze how users are engaging with the email service.

Approach:

The database has been created using MySQL. The database contains information on multiple tables about various fields. SQL queries are the used to extract the required data from the tables in order to acquire the corresponding insights.

Tech-Stack Used:

Software & Version used: MySQL Workbench 8.0 CE

Purpose: Stores data, Manages the databases and supports the SQL commands to analyze the results.

Insights:

For Case Study 1: Job Data Analysis

A) Jobs Reviewed Over Time:

```

select ds as date, round( ( count(job_id)/sum(time_spent) ) * 3600 )
as 'Jobs Reviewed'
from job_data
where ds between '11/01/2020' and '11/30/2020'
group by ds;

```

date	Jobs Reviewed
11/30/2020	180
11/29/2020	180
11/28/2020	218
11/27/2020	35
11/26/2020	64
11/25/2020	80

Inference: Hence, these are the number of jobs reviewed per hour for each day in November 2020.

B) Throughput Analysis:

```

select ds, count(job_id) as 'num jobs',
sum(time_spent) as 'overall time spent',
count(job_id)/sum(time_spent) as throughput,
truncate( avg(job_id)/avg(time_spent),3 ) as '7d rolling avg throughput'
from job_data
group by ds
order by ds;

```

ds	num jobs	overall time spent	throughput	7d rolling avg throughput
11/25/2020	1	45	0.0222	0.444
11/26/2020	1	56	0.0179	0.41
11/27/2020	1	104	0.0096	0.105
11/28/2020	2	33	0.0606	1.454
11/29/2020	1	20	0.05	1.15
11/30/2020	2	40	0.05	1.075

Inference: 7-day rolling average for throughput is preferred which is useful for long term trends analysis.

C) Language Share Analysis:

```

select language, (count(*)/(select count(*) from job_data)) * 100
as Percentage_share from job_data
group by language;

```

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

Inference: These are the percentage share of each language in the last 30 days and Persian has been shared the most percentage.

A) Duplicate Rows Detection:

```
Select count(*), ds, job_id, actor_id from job_data
group by ds, job_id, actor_id
having count(*) > 1;
```

count(*)	ds	job_id	actor_id
----------	----	--------	----------

Inference: These are no duplicate rows found from the data.

For Case Study 2: Investigating Metric Spike

A) Weekly User Engagement:

```
select (select count(distinct(user_id)) as 'active users' from users
where state = 'active')/(select count(distinct(user_id)) as 'active users'
from users)*100 as 'weekly user engagement'
from users;
```

weekly user engagement
49.2

Inference: Hence, the activeness of users on a weekly basis has been calculated which is 49.2%

B) User Growth Analysis:

```
select newusers.device, avg(newusers.newusers/1st_month.overall_users*100) as 'user
growth over time for a product'
from (select device, month(occurred_at) as signup_month,
count(distinct user_id) as newusers from events
group by device, signup_month) as newusers
join (select device, count(distinct user_id) as overall_users from events
group by device) as 1st_month
on newusers.device = 1st_month.device
group by newusers.device;
```

device	user growth over time for a product
acer aspire desktop	36.49
acer aspire notebook	35.58
amazon fire phone	32.2
asus chromebook	34.89
dell inspiron desktop	35.56
dell inspiron notebook	36.38
hp pavilion desktop	35.3
htc one	33.37
ipad air	33.14
ipad mini	32.34
iphone 4s	34.7
iphone 5	35.11
iphone 5s	33.99
kindle fire	32.64
lenovo thinkpad	36.15
mac mini	39.41
macbook air	35.63
macbook pro	35.86
nexus 10	33
nexus 5	34.28
nexus 7	32.81
nokia lumia 635	36.25
samsung galaxy tablet	31.85
samsung galaxy note	33.67
samsung galaxy s4	33.64
windows surface	32.35

Inference: Here are the user growth over time for all the products which is more than 30%.

C) Weekly Retention Analysis:

```

Select WEEK(occurred_at) AS week,
count(distinct user_id) as 'sign-up cohort',
count(distinct CASE when event_type='engagement' then user_id END) as engagement,
count(distinct CASE when event_type='engagement' then user_id END)/count(distinct
user_id)*100 as 'retention of users' from events
group by WEEK(occurred_at)
order by week;
```

week	sign-up cohort	engagement	retention of users
17	740	663	89.5946
18	1260	1068	84.7619
19	1287	1113	86.4802
20	1351	1154	85.4182
21	1299	1121	86.2972
22	1381	1186	85.8798
23	1446	1232	85.2006
24	1471	1275	86.6757
25	1459	1264	86.6347
26	1509	1302	86.2823
27	1573	1372	87.2219
28	1577	1365	86.5568
29	1607	1376	85.6254
30	1706	1467	85.9906
31	1514	1299	85.7992
32	1454	1225	84.2503
33	1438	1225	85.1878
34	1443	1204	83.4373
35	118	104	88.1356

Inference: Here are the weekly retention of users based on their sign-up cohort. It is analyzed that the retention rate is more than 80%.

D) Weekly Engagement Per Device:

```
select WEEK(occurred_at) as week,
count(distinct CASE when event_type = 'engagement' then user_id END)/count(distinct
user_id)/(select count(distinct(device)) from events)*100
as 'weekly Engagement rate per device' from events
group by WEEK(occurred_at)
order by week;
```

week	weekly Engagement rate per device	week	weekly Engagement rate per device
17	3.44594594	27	3.35468727
18	3.26007326	28	3.3291059
19	3.32616102	29	3.29328419
20	3.28531572	30	3.30733159
21	3.31912122	31	3.29996951
22	3.30306912	32	3.24039784
23	3.27694435	33	3.27645233
24	3.33368195	34	3.20912628
25	3.33210312	35	3.38983051
26	3.31855023		

Inference: Hence, the activeness of the users on a weekly basis per device has been calculated.

E) Email Engagement Analysis:

```
select email_engagements, overall_users,  
email_engagements/overall_users*100 as "engagement metrics rate"  
from (select count(distinct(user_id)) as overall_users,  
count(distinct CASE  
when action = 'email_open' then user_id  
when action = 'email_clickthrough' then user_id END) as email_engagements  
from email_events) as counts
```

email_engagements	overall_users	engagement metrics rate
5927	6179	95.9217

Inference: The email engagement metrics has been calculated and the users are engaging better.

Result:

The 'Operation Analytics and Investigating Metric Spike' data is Analyzed and the insights are obtained, that can be used by the operations, support, and marketing teams to identify the areas for improvement within the company. Based on the findings, sudden changes in key metrics such as User engagement, User's growth, email engagement and so on have been investigated in order to improve the platform. Finally, contributing to and improving the overall experience can help the company grow.