

Operation Analytics and Investigating Metric Spike

Project Description : It is a project related to Operation Analytics. 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. In this project we have analysed job reviews, no of events happened in company, how many languages speak in company and analyse duplication record. 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. We have lots of questions regarding engagement, growth, weekly retention, email etc.

Approach : first we analyse the dataset which is provided by company. We must check duplication entry, table records, data etc. Investigating Metric Spike, we have three table for users, events, and email events.

Tech-Stack Used : we have excel file using excel file we have create table using SQL query on online portal(db-fiddle.com). For case study two we are used Mode studio for tables.

Insights :

Case Study 1 (Job Data):

Number of jobs reviewed : Amount of jobs reviewed over time.

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

Query : `SELECT COUNT(DISTINCT job_id)/(30*24) FROM job_data WHERE
ds BETWEEN '2020-11-01' AND '2020-11-30'`

Output :

| |
|---|
| <code>count(distinct job_id)/(30*24)</code> |
| 0.0111 |

Throughput : It is the no. of events happening per second.

Task : Let us 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?

Query :

```
SELECT ds,jobs_reviewed,AVG(jobs_reviewed) OVER (ORDER BY ds rows BETWEEN 6
preceding AND current row) AS throughput_7 FROM
(SELECT ds,COUNT(DISTINCT job_id) AS jobs_reviewed FROM job_data WHERE ds BETWEEN
'2020-11-01' AND '2020-11-30'
GROUP BY ds ORDER BY ds)a
```

Output :

Results

Query #1 Execution time: 1ms

| ds | jobs_reviewed | throughput |
|------------|---------------|------------|
| 2020-11-25 | 1 | 1.0000 |
| 2020-11-26 | 1 | 1.0000 |
| 2020-11-27 | 1 | 1.0000 |
| 2020-11-28 | 2 | 1.2500 |
| 2020-11-29 | 1 | 1.2000 |
| 2020-11-30 | 2 | 1.3333 |

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?

Query : SELECT language, num_jobs,100.0*num_jobs/total_jobs AS pct_jobs FROM
(SELECT language, COUNT (DISTINCT job_id) AS num_jobs FROM job_data GROUP BY language)
a

CROSS JOIN

(SELECT COUNT (DISTINCT job_id) AS total_job FROM job_data)b

Output :

Query #1 Execution time: 1ms

| language | num_jobs | pct_jobs |
|----------|----------|----------|
| Arabic | 1 | 12.50000 |
| English | 1 | 12.50000 |
| French | 1 | 12.50000 |
| Hindi | 1 | 12.50000 |
| Italian | 1 | 12.50000 |
| Persian | 3 | 37.50000 |

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

Task : Let us say you see some duplicate rows in the data. How will you display duplicates from the table?

Query : SELECT * FROM

(SELECT *, ROW_NUMBER() OVER(PARTITION BY job_id) AS rownum

FROM job_data) a

WHERE rownum > 1

Output :

Query #1 Execution time: 2ms

| job_id | actors_id | event | language | time_spent | org | ds | rownum |
|--------|-----------|----------|----------|------------|-----|------------|--------|
| 28 | 1008 | transfer | Italian | 45 | C | 2020-11-25 | 2 |

Case Study 2 (Investigating metric spike) :

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?

Query : SELECT EXTRACT(week FROM occurred_at) AS weeknum , COUNT(DISTINCT user_id)
FROM tutorial.yammer_events a GROUP BY weeknum

Output:

| | weeknum | count |
|----|---------|-------|
| 1 | 18 | 791 |
| 2 | 19 | 1244 |
| 3 | 20 | 1270 |
| 4 | 21 | 1341 |
| 5 | 22 | 1293 |
| 6 | 23 | 1366 |
| 7 | 24 | 1434 |
| 8 | 25 | 1462 |
| 9 | 26 | 1443 |
| 10 | 27 | 1477 |
| 11 | 28 | 1556 |
| 12 | 29 | 1556 |
| 13 | 30 | 1593 |
| 14 | 31 | 1685 |
| 15 | 32 | 1483 |
| 16 | 33 | 1438 |
| 17 | 34 | 1412 |
| 18 | 35 | 1442 |

User Growth : Amount of users growing over time for a product.

Task : Calculate the user growth for product?

Query : SELECT year, weeknum, num_active_user, SUM(num_active_user)OVER(ORDER BY year,weeknum ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cum_active_users FROM (SELECT EXTRACT(year from a.activated_at) AS year, EXTRACT(week from a.activated_at) AS weeknum, COUNT(DISTINCT user_id) AS num_active_user FROM tutorial.yammer_users a WHERE state='active' GROUP BY year, weeknum) a

Output :

| | year | weeknum | num_active_user | cum_active_users | |
|----|------|---------|-----------------|------------------|--|
| 1 | 2013 | 1 | 67 | 67 | |
| 2 | 2013 | 2 | 29 | 96 | |
| 3 | 2013 | 3 | 47 | 143 | |
| 4 | 2013 | 4 | 36 | 179 | |
| 5 | 2013 | 5 | 30 | 209 | |
| 6 | 2013 | 6 | 48 | 257 | |
| 7 | 2013 | 7 | 41 | 298 | |
| 8 | 2013 | 8 | 39 | 337 | |
| 9 | 2013 | 9 | 33 | 370 | |
| 10 | 2013 | 10 | 43 | 413 | |
| 11 | 2013 | 11 | 33 | 446 | |
| 12 | 2013 | 12 | 32 | 478 | |
| 13 | 2013 | 13 | 33 | 511 | |
| 14 | 2013 | 14 | 40 | 551 | |
| 15 | 2013 | 15 | 35 | 586 | |
| 16 | 2013 | 16 | 42 | 628 | |
| 17 | 2013 | 17 | 48 | 676 | |
| 18 | 2013 | 18 | 48 | 724 | |

Weekly Retention : Users getting retained weekly after signing-up for a product.

Task : Calculate the weekly retention of users-sign up cohort?

Query : SELECT COUNT(user_id), SUM(CASE WHEN retention_week = 1 THEN 1 ELSE 0 END) AS week_1 FROM (SELECT a.user_id, a.signup_week, b.engagement_week, b.engagement_week - a.signup_week AS retention_week FROM ((SELECT DISTINCT user_id, EXTRACT(week FROM occurred_at) AS signup_week FROM tutorial.yammer_events WHERE event_type = 'signup_flow' AND event_name = 'complete_signup' AND EXTRACT(week FROM occurred_at) = 18) a LEFT JOIN (SELECT DISTINCT user_id, EXTRACT(week FROM occurred_at) AS engagement_week FROM tutorial.yammer_events WHERE event_type = 'engagement') b ON a.user_id = b.user_id) ORDER BY a.user_id) a

Output :

| | count | week_1 |
|---|-------|--------|
| 1 | 317 | 64 |

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?

Query : SELECT EXTRACT(year FROM occurred_at) AS year, EXTRACT(week FROM occurred_at) AS week, device, COUNT(DISTINCT user_id) FROM tutorial.yammer_events WHERE event_type='engagement' GROUP BY 1,2,3 ORDER by 1,2,3

Output :

| | year | week | device | count |
|----|------|------|------------------------|-------|
| 1 | 2014 | 18 | acer aspire desktop | 10 |
| 2 | 2014 | 18 | acer aspire notebook | 21 |
| 3 | 2014 | 18 | amazon fire phone | 4 |
| 4 | 2014 | 18 | asus chromebook | 23 |
| 5 | 2014 | 18 | dell inspiron desktop | 21 |
| 6 | 2014 | 18 | dell inspiron notebook | 49 |
| 7 | 2014 | 18 | hp pavilion desktop | 15 |
| 8 | 2014 | 18 | htc one | 16 |
| 9 | 2014 | 18 | ipad air | 30 |
| 10 | 2014 | 18 | ipad mini | 21 |
| 11 | 2014 | 18 | iphone 4s | 21 |
| 12 | 2014 | 18 | iphone 5 | 70 |
| 13 | 2014 | 18 | iphone 5s | 45 |
| 14 | 2014 | 18 | kindle fire | 6 |
| 15 | 2014 | 18 | lenovo thinkpad | 90 |
| 16 | 2014 | 18 | macbook air | 57 |
| 17 | 2014 | 18 | macbook pro | 154 |
| 18 | 2014 | 18 | mac mini | 8 |

Email Engagement : Users engaging with the email service.

Task : Calculate the email engagement metrics?

Query : SELECT 100.0 *SUM(CASE WHEN email_cat = 'email_open' THEN 1 ELSE 0 END)/SUM(CASE WHEN email_cat = 'email_sent' THEN 1 ELSE 0 END) AS email_open_rate, 100.0 *SUM(CASE WHEN email_cat = 'email_clicked' THEN 1 ELSE 0 END)/SUM(CASE WHEN email_cat = 'email_sent' THEN 1 ELSE 0 END) AS email_clicked_rate FROM (SELECT *, CASE WHEN action IN ('sent_weekly_digest', 'sent_reengagement_email') THEN 'email_sent' WHEN action IN ('email_open') THEN 'email_open' WHEN action in ('email_clickthrough') THEN 'email_clicked' END AS email_cat FROM tutorial.yammer_emails) a

Output :

| | email_open_rate | email_clicked_rate |
|---|-----------------|--------------------|
| 1 | 33.5834 | 14.7899 |

Result : In first case study we have calculated the number of jobs reviewed, throughput, share percentage of language in company and duplicate row.

In second case study we are calculated the growth , engagement, retention, email attention etc using SQL query.