Operation Analytics and Investigating Metric Spike

08.03.2023
—

Mowleeshwaran V R
Chennai,
Tamil Nadu,603 203

Description

The Operation Analytics and Investigating Metric Spike project aims to identify and investigate anomalies and spikes in operational metrics of a business or organization. The project seeks to leverage data analytics and visualization tools to identify these anomalies, diagnose their root cause, and provide insights that can inform decision-making and optimize business operations.

Approach

The approach to the Operation Analytics and Investigating Metric Spike project will involve the following steps:

1. Define operational metrics: The first step will be to define the operational metrics that the organization wants to track and monitor. This may include metrics such as website traffic, sales revenue, customer acquisition rate, customer churn rate, or any other key performance indicators (KPIs) that are critical to the organization's success.

2. Collect data: Once the operational metrics have been identified, the next step will be to collect relevant data from various sources such as internal databases, web analytics tools, social media platforms, and other sources.

3. Clean and preprocess data: Before conducting any analysis, the data will need to be cleaned and preprocessed to ensure accuracy and completeness. This may involve removing missing data, correcting errors, or normalizing data across different sources.

4. Analyze data: The data will be analyzed using various statistical and data analytics techniques to identify any anomalies or spikes in the operational metrics. This may involve using time series analysis, regression analysis, or other statistical methods to identify trends, patterns, or outliers in the data.

5. Visualize data: The insights derived from the analysis will be visualized using data visualization tools such as charts, graphs, or dashboards to provide a clear and concise overview of the operational metrics and any identified anomalies or spikes.

6. Investigate root cause: Once anomalies or spikes have been identified, the next step will be to investigate their root cause. This may involve conducting further analysis or consulting with subject matter experts to identify any external or internal factors that may have contributed to the anomaly.

7. Provide insights and recommendations: Based on the analysis and investigation, the project team will provide insights and recommendations to the organization on how to optimize their operations and improve performance. This may involve identifying areas for improvement, developing new strategies or tactics, or optimizing existing processes to prevent future anomalies or spikes.

Tech Stack Used:

MySQL Workbench 8.0 CE(Provides a visual console to easily administer MySQL environments and gain better visibility into databases).

Analyze:

Case Study 1 (Job Data)

Below is the structure of the table with the definition of each column that you must work on:

Table-1: job_data

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

Use the dataset attached in the Dataset section below the project images then answer the questions that follows

Tasks:

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

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

SQL QUERY:SELECT (count(job_id))/(30*24) AS NO_OF_JOBS_REVIEWED spike.datao;

SQL QUERY:SELECT (count(distinct(job_id)))/(30*24) AS NO_OF_DISTINCT_JOBS_REVIEWED from spike.datao;

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

Your task: 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:SELECT ds as Date, Jobs_Reviewed, AVG(Jobs_Reviewed)OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) AS ThroughtPut_7_Rolling_Average FROM (SELECT ds, COUNT(DISTINCT job_id) AS jobs_reviewed FROM spike.datao GROUP BY ds ORDER BY ds) a;

3.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:select datao.language,count(distinct datao.language) as num_jobs,((count(datao.language)/(select count(*) from datao))*100) as percnt_share from datao group by datao.language;

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

Your task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

SQL QUERY:SELECT * FROM(SELECT *, ROW_NUMBER()OVER(PARTITION BY job_id) AS row_num FROM spike.datao) a WHERE row_num>1;

Case Study 2 (Investigating metric spike)

The structure of the table with the definition of each column that you must work on is present in the project image

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.

Use the dataset attached in the Dataset section below the project images then answer the questions that follows

Tasks:

1.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:SELECT extract(week from created_at) as week_num_user,count(distinct user_id) as No_of_users FROM spike.userdet group by week_num_user;

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

Your task: Calculate the user growth for the product?

SQL QUERY:select No_of_Years,No_of_Weeks,Num_Of_Active_Users, SUM(Num_Of_Active_Users)OVER(ORDER BY No_Of_Years,No_of_Weeks ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS Sum_of_Active_Users from (select extract(year from activated_at) as No_of_years,extract(week from activated_at) as No_of_Weeks,count(distinct user_id) as Num_Of_Active_Users from spike.userdet WHERE state = 'active' group by No_of_Years,No_of_Weeks order by No_of_Years,No_of_Weeks) a;

3.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 distinct user_id,COUNT(user_id),SUM(CASE WHEN retention_week = 1 Then 1 Else 0 END) as per_week_retention FROM (SELECT first.user_id,first.signup_week, second.engagement_week,second.engagement_week - first.signup_week as retention_week FROM((SELECT distinct(user_id), extract(week from occurred_at) as signup_week from spike.event WHERE event_type = 'signup_flow' and event_name = 'complete_signup' --and extract(week from occurred_at) = 18) LEFT JOIN (SELECT distinct user_id, extract (week from occurred_at) as engagement_week FROM spike.event where event_type = 'engagement') on first.user_id = second.user_id)) group by user_id order by user_id;

4.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 extract(year from occurred_at) as year_num, extract(week from occurred_at) as week_num,device,COUNT(distinct user_id) as no_of_users FROM spike.event where event_type = 'engagement' group by 1,2,3 order by 1,2,3;

5.Email Engagement: Users engaging with the email service.

Your task: Calculate the email engagement metrics?

SQL QUERY:SELECT 100.0*SUM(CASE when email_cat = 'email_opened' then 1 else 0 end)/SUM(CASE when email_cat = 'email_sent' then 1 else 0 end) as email_opening_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_clicking_rate FROM(SELECT *,CASE WHEN action in ('sent_weekly_digest','sent_reengagement_email') then 'email_sent' WHEN action in ('email_open') then 'email_opened' WHEN action in ('email_clickthrough') then 'email_clicked' end as email_cat from spike.emails)a;

Insights:

Case Study 1 (Job Data) Tasks:

1.Number of jobs reviewed:

NO_OF_DISTINCT_JOBS_REVIEWED
0.0083

NO_OF_JOBS_REVIEWED
0.0111

2.Throughput:

Date	Jobs_Reviewed	ThroughtPut_7_Rolling_Average
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

3.Percentage share of each language:

language	num_jobs	percnt_share
English	1	12.5000
Arabic	1	12.5000
Persian	3	37.5000
Hindi	1	12.5000
French	1	12.5000
Italian	1	12.5000

4.Duplicate rows:

	ds	job_id	actor_id	event	language	time_spent	org	row_num
▶	2020-11-28	23	1005	transfer	Persian	22	D	2
	2020-11-26	23	1004	skip	Persian	56	A	3

Case Study 2 (Investigating metric spike)Tasks:

1.User Engagement:

https://drive.google.com/file/d/1Bwy6162Vp-0uVAe01aa_0jIUC1K8i48I/view?usp=sharing

2.User Growth:

https://drive.google.com/file/d/1_5J-PK7BNUsYwPj8h49p0WhEO03rSZur/view?usp=sharing

3.Weekly Retention:
https://drive.google.com/file/d/1u--A5sQKG2kW9m1F_zbzB7OlfodSRYrQ/view?usp=sharing

4.Weekly Engagement:

https://drive.google.com/file/d/1ibuobqvWRcpTLgM0GDzZanV_bVsnR8wK/view?usp=share_link

5.Email Engagement:

https://drive.google.com/file/d/1X6TB2Y_t2NWTuDdXSRrT4IpZYskz6w7W/view?usp=sharing

Result:

Overall, by implementing these recommendations, the organization can improve its ability to identify, analyze, and respond to metric spikes, leading to more efficient and effective operations.