

# **OPERATION ANALYTICS & INVESTIGATING METRIC**

## **SPIKE CASE STUDY**

# CONTENT

- 01** PROJECT DESCRIPTION
- 02** APPROACH
- 03** TECH-STACK USED
- 04** INSIGHTS
- 05** RESULT

# PROJECT DESCRIPTION

The project "Operation Analytics & Investigating Metric Spike" focuses on analyzing end-to-end operations using data-driven insights to drive growth and efficiency. As a Data Analyst, you collaborate with teams like operations, support, and marketing to extract valuable insights from collected data. Operation analytics is crucial for identifying bottlenecks, streamlining processes, and improving cross-functional collaboration. It enables informed decision-making, automation, and effective workflows, shaping a company's success. Investigating metric spikes is an essential aspect, aiding in understanding fluctuations in key metrics and facilitating proactive problem-solving.

# APPROACH

Using SQL queries, I will calculate the number of jobs reviewed per hour per day for November 2020, analyze the 7-day rolling average of throughput, determine the percentage share of each language in the last 30 days, and identify and display duplicate rows from the table. This data-driven approach will provide valuable insights, trends, and patterns for informed decision-making.

For the "Operation Analytics & Investigating Metric Spike" project, I adopted a data-driven approach to extract and analyze relevant information from the company's database. Using a combination of SQL queries and data analysis techniques, I retrieved the necessary data to investigate metric spikes and derive insights.

Overall, through the application of SQL queries, data analysis techniques, and informative reporting, I was able to provide valuable insights into the company's operations, investigate metric spikes, and support data-driven decision-making processes.

In this project, I will analyze user activity data to calculate weekly user engagement, track user growth over time, determine the weekly retention of user sign-up cohorts, measure weekly engagement per device, and evaluate email engagement metrics.

These analyses will provide valuable insights into user satisfaction, product/service growth, user retention, device-specific engagement, and the effectiveness of email communication.

# TECH-STACK USED

In the "Operation Analytics & Investigating Metric Spike" project, I utilized SQL queries and MySQL as the primary tool for data extraction and analysis. Leveraging the power of MySQL, I efficiently retrieved and queried relevant data from the company's database. By applying various analysis techniques, I gained valuable insights into operational performance, identified areas for improvement, and investigated metric spikes. This data-driven approach facilitated informed decision-making and optimization of workflows, ultimately leading to enhanced efficiency and cross-functional collaboration within the company. The combination of SQL queries and MySQL proved to be instrumental in driving operational analytics and investigating metric spikes for improved overall performance.

# INSIGHTS - JOB DATA

```
mysql> SELECT DATE(ds) AS date,
-> COUNT(DISTINCT job_id) AS jobs_reviewed
-> FROM job_data
-> WHERE ds >= '2020-11-01' AND ds < '2020-12-01'
-> GROUP BY date
-> ORDER BY date;
+-----+-----+
| date      | jobs_reviewed |
+-----+-----+
| 2020-11-25 |           1 |
| 2020-11-26 |           1 |
| 2020-11-27 |           1 |
| 2020-11-28 |           2 |
| 2020-11-29 |           1 |
| 2020-11-30 |           2 |
+-----+-----+
6 rows in set (0.00 sec)
```

**No of jobs reviewed:** In November 2020, a total of 8 jobs were reviewed. Looking at the data, there seems to be a consistent trend of reviewing 1 job per day, with a few instances of higher activity. Specifically, on November 28th and 30th, there was an increased number of job reviews with 2 reviews each day. This suggests a potential spike in activity or a specific event that prompted more job reviews during those days. Further analysis of the underlying factors and correlation with other metrics could provide valuable insights into the reasons behind these spikes and help identify opportunities for improvement in job review processes.

# INSIGHTS - JOB DATA

```
mysql> SELECT t1.ds, t1.throughput_per_day, AVG(t2.throughput_per_day) AS rolling_average
--> FROM (
-->     SELECT ds, COUNT(*) / SUM(time_spent) AS throughput_per_day
-->     FROM job_data
-->     GROUP BY ds
--> ) AS t1
--> JOIN (
-->     SELECT ds, COUNT(*) / SUM(time_spent) AS throughput_per_day
-->     FROM job_data
-->     GROUP BY ds
--> ) AS t2 ON t1.ds >= t2.ds AND t1.ds <= DATE_ADD(t2.ds, INTERVAL 6 DAY)
--> GROUP BY t1.ds, t1.throughput_per_day;
+-----+-----+-----+
| ds      | throughput_per_day | rolling_average |
+-----+-----+-----+
| 2020-11-30 |      0.0500 |      0.03505000 |
| 2020-11-29 |      0.0500 |      0.03206000 |
| 2020-11-28 |      0.0606 |      0.02757500 |
| 2020-11-27 |      0.0096 |      0.01656667 |
| 2020-11-26 |      0.0179 |      0.02005000 |
| 2020-11-25 |      0.0222 |      0.02220000 |
+-----+-----+-----+
6 rows in set (0.01 sec)
```

# INSIGHTS - JOB DATA

**Throughput:** The 7-day rolling average of throughput reveals a gradual decline in the rate of events happening per second. On November 30th, the throughput was 0.035 events per second, slightly lower than the previous day's average of 0.032 events per second. This trend suggests a potential need for optimization and improvement to enhance event processing and system efficiency.

The preference for either the daily metric or the 7-day rolling average when analyzing throughput depends on the analysis objectives. The daily metric offers a quick snapshot and facilitates the detection of short-term trends and anomalies, while the 7-day rolling average provides a smoothed representation of the overall trend, capturing longer-term patterns and changes in performance. Choosing between the two depends on whether the focus is on immediate fluctuations or gaining a broader perspective on throughput patterns. Aligning the choice with the analysis objectives ensures the most meaningful insights are derived from the throughput data.

# INSIGHTS - JOB DATA

```
mysql> SELECT language, COUNT(*) / total_events * 100 AS percentage_share
-> FROM (
->   SELECT language, COUNT(*) AS total_events
->   FROM job_data
->   WHERE ds >= DATE_SUB(CURDATE(), INTERVAL 30 DAY) AND ds <= CURDATE()
->   GROUP BY language
-> ) AS subquery
-> GROUP BY language;
Empty set (0.00 sec)
```

**Percentage Share of each language:** The data for the last 30 days does not provide any information on the percentage share of each language for different contents. This indicates that there is no available data or records to analyze in order to determine the language distribution across different content types. Further investigation or data collection may be necessary to gain insights into the language distribution and its impact on content diversity or user preferences.

# INSIGHTS - JOB DATA

```
mysql> SELECT *  
-> FROM job_data  
-> WHERE (job_id, actor_id, event, language, time_spent, org, ds) IN (  
->     SELECT job_id, actor_id, event, language, time_spent, org, ds  
->     FROM job_data  
->     GROUP BY job_id, actor_id, event, language, time_spent, org, ds  
->     HAVING COUNT(*) > 1  
-> );  
Empty set (0.01 sec)
```

**Duplicate Rows:** The absence of duplicate rows in the data indicates that there are no instances where rows have the exact same values. This suggests a level of data integrity and uniqueness within the table. The lack of duplicates simplifies data analysis and ensures that each row represents distinct information.

- To display duplicate rows from a table in MySQL, you can use a query with the `SELECT` statement and the `DISTINCT` keyword. The query compares columns in the table using a subquery with the `GROUP BY` clause and the `COUNT(\*)` function to identify rows with duplicate values. The query then retrieves the duplicate rows based on the selected column values.

# INSIGHTS - INVESTIGATING METRIC SPIKE

```
mysql> SELECT WEEK(occurred_at) AS week_number, COUNT(DISTINCT user_id) AS weekly_user_engagement  
-> FROM Events  
-> GROUP BY week_number;  
+-----+-----+  
| week_number | weekly_user_engagement |  
+-----+-----+  
| 17          |        740           |  
| 18          |       1260           |  
| 19          |       1287           |  
| 20          |       1351           |  
| 21          |       1299           |  
| 22          |       1381           |  
| 23          |       1446           |  
| 24          |       1471           |  
| 25          |       1459           |  
| 26          |       1509           |  
| 27          |       1573           |  
| 28          |       1577           |  
| 29          |       1607           |  
| 30          |       1706           |  
| 31          |       1514           |  
| 32          |       1454           |  
| 33          |       1438           |  
| 34          |       1443           |  
| 35          |        118           |  
+-----+-----+  
19 rows in set (0.69 sec)
```

**User Engagement:** The data shows a generally increasing trend in weekly user engagement, reaching a peak of 1706 engagements in week 30. However, there is a noticeable decline in engagement in week 35 with only 118 engagements recorded. Further investigation is needed to understand the reasons behind this drop and ensure continued user satisfaction and engagement.

# INSIGHTS - INVESTIGATING METRIC SPIKE

```
mysql> SELECT WEEK(created_at) AS week_number, COUNT(DISTINCT user_id) AS user_growth
    -> FROM Users
    -> GROUP BY week_number;
+-----+-----+
| week_number | user_growth |
+-----+-----+
|          0 |        197 |
|          1 |        300 |
|          2 |        299 |
|          . |        .   |
|          . |        .   |
|          . |        .   |
|          50 |        221 |
|          51 |        235 |
|          52 |        87 |
+-----+-----+
53 rows in set (0.04 sec)
```

**User Growth:** The data shows a generally increasing trend in user growth over time, with a peak of 618 new users in week 30. However, there is a noticeable decline in week 35, with only 196 new users recorded. Further investigation is needed to understand the factors contributing to this decline and take necessary measures to ensure sustained user growth.

# INSIGHTS - INVESTIGATING METRIC SPIKE

```
mysql> SELECT WEEK(u.created_at) AS week_number, COUNT(DISTINCT e.user_id) AS weekly_retention
-> FROM Users u
-> JOIN Events e ON u.user_id = e.user_id
-> WHERE WEEK(e.occurred_at) >= WEEK(u.created_at)
-> GROUP BY week_number;
+-----+-----+
| week_number | weekly_retention |
+-----+-----+
| 0           | 37              |
| 1           | 54              |
| 2           | 61              |
| 3           | 55              |
| :           | :               |
| :           | :               |
| 33          | 485             |
| 34          | 501             |
| 35          | 33              |
+-----+-----+
36 rows in set (0.34 sec)
```

**Weekly Retention:** The weekly retention of the user sign-up cohort demonstrates a generally positive trend, with fluctuations. Retention rates increase over time, reaching a peak of 492 users in week 30. However, there is a significant decline in week 35, with only 33 users retained. Further analysis is needed to understand the cause of this drop and implement strategies to improve user retention.

# INSIGHTS - INVESTIGATING METRIC SPIKE

```
mysql> SELECT device, COUNT(*) AS total_engagement, COUNT(DISTINCT WEEK(occurred_at)) AS occurrences,
->           COUNT(*) / COUNT(DISTINCT WEEK(occurred_at)) AS weekly_engagement
->     FROM events
->    WHERE device IN ('lenovo thinkpad', 'samsung galaxy s4', 'dell inspiron desktop', 'nexus 7', 'macbook pro',
->                      'iphone 5s', 'iphone 4s', 'ipad air', 'iphone 5', 'kindle fire', 'asus chromebook',
->                      'nexus 5', 'hp pavilion desktop', 'macbook air', 'acer aspire notebook', 'htc one',
->                      'mac mini', 'amazon fire phone', 'acer aspire desktop', 'dell inspiron notebook',
->                      'nokia lumia 635', 'ipad mini', 'nexus 10', 'windows surface', 'samsumg galaxy tablet',
->                      'samsung galaxy note')
->   GROUP BY device;
+-----+-----+-----+
| device          | total_engagement | occurrences | weekly_engagement |
+-----+-----+-----+
| acer aspire desktop |      5402 |        19 |       284.3158 |
| acer aspire notebook |     9372 |        19 |       493.2632 |
| amazon fire phone |     2298 |        19 |       120.9474 |
| asus chromebook |    10013 |        19 |       527.0000 |
| dell inspiron desktop |    10569 |        19 |       556.2632 |
| dell inspiron notebook |   20476 |        19 |      1077.6842 |
| hp pavilion desktop |    9280 |        19 |       488.4211 |
| htc one |        4485 |        19 |       236.0526 |
| ipad air |       9994 |        19 |       526.0000 |
| ipad mini |      5895 |        19 |       310.2632 |
| iphone 4s |     10097 |        19 |       531.4211 |
| iphone 5 |     27134 |        19 |      1428.1053 |
| iphone 5s |     16703 |        19 |       879.1053 |
| kindle fire |      4279 |        19 |       225.2105 |
| lenovo thinkpad |    38679 |        19 |      2035.7368 |
| mac mini |      4621 |        19 |       243.2105 |
| macbook air |     28103 |        19 |      1479.1053 |
| macbook pro |     59946 |        19 |      3155.0526 |
| nexus 10 |      5446 |        19 |       286.6316 |
| nexus 5 |     17249 |        19 |       907.8421 |
| nexus 7 |      6894 |        19 |       362.8421 |
| nokia lumia 635 |    5880 |        19 |       309.4737 |
| samsumg galaxy tablet |   1920 |        18 |      106.6667 |
| samsung galaxy note |    2821 |        19 |       148.4737 |
| samsung galaxy s4 |    19594 |        19 |      1031.2632 |
| windows surface |    3673 |        19 |       193.3158 |
+-----+-----+-----+
```

**Weekly Engagement:** The data reveals varying levels of weekly engagement per device. The Macbook Pro and Lenovo Thinkpad exhibit high engagement, while the Samsung Galaxy Tablet and Amazon Fire Phone show relatively lower engagement. These insights highlight devices with strong user satisfaction and areas that may require improvement.

# INSIGHTS - INVESTIGATING METRIC SPIKE

```
mysql> SELECT action, COUNT(*) AS email_engagement
-> FROM Email_events
-> GROUP BY action;
+-----+-----+
| action | email_engagement |
+-----+-----+
| sent_weekly_digest | 57267 |
| email_open | 20459 |
| email_clickthrough | 9010 |
| sent_reengagement_email | 3653 |
+-----+-----+
4 rows in set (0.12 sec)
```

**Email Engagement:** The email engagement metrics indicate that a significant number of users engaged with the email service, with 57,267 receiving the weekly digest, 20,459 opening the emails, 9,010 clicking through, and 3,653 receiving reengagement emails.

# RESULT

During the process of making this project, I have achieved several significant milestones. Firstly, I gained hands-on experience in working with a real-world dataset and using SQL queries to extract meaningful information. This allowed me to develop strong data analysis and querying skills. Additionally, I learned how to interpret the insights derived from the data, such as user engagement patterns, popular hashtags, and identifying potential fake accounts. This project has helped me strengthen my analytical thinking, problem-solving, and presentation skills. It has provided me with practical knowledge in the field of data analysis and has boosted my confidence in handling similar projects in the future.

# THANK YOU!

PRESENTED BY: MIHIR PATEL

