Anushka Gharage

PROJECT-3

Operation Analytics and Investigating Metric Spike

Project Description

The project aimed to analyze operational data from the 'job_data' table to derive meaningful insights for business improvement. The purpose was to understand user behaviour, identify trends, and pinpoint areas for optimization within the company's operations. The analysis focused on metrics like job reviews, event types, languages, and time spent, offering a comprehensive view of user interactions.
The project also aimed to analyze user engagement data from various sources, including events, emails, and user details, to gain insights into user behaviour. The purpose was to understand how users are engaging with the product and email services, optimize email campaigns, and enhance overall user experience.

Approach

1. Data Gathering & Understanding: Collected data from the 'job_data', 'events', 'users', and 'email_events' tables, understanding their structures and relationships, including its columns and their meanings, ensuring clarity on the data's structure and content.
2. Data Cleaning: Identified and handled missing values in Case Study 2 tables.
3. Data Analysis: Utilized SQL queries to calculate various metrics like job reviews per hour, language shares, and event throughputs, facilitating in-depth analysis, calculate engagement metrics, including user activity trends, email open rates, and device-specific engagement.

Tech-Stack Used

MySQL Workbench: Utilized for SQL query writing, data exploration and analysis.

Insights

- **CASE STUDY 1:**
    1. Peak Activity Hours: Identified peak hours of job reviews, assisting in resource allocation and task scheduling.
    2. Language Preferences: Determined user language preferences, guiding content localization efforts and user experience improvements.
    3. Even Throughput: Analyzed throughput for various events, aiding in understanding user engagement patterns and platform usage.

4. Identified Duplicates: Discovered and dealt with duplicate entries, ensuring data accuracy for future analyses.

- **CASE STUDY 2:**
    1. User Engagement Patterns: Analyzed user engagement trends over time, identifying peak activity hours and days. Observed that weekends had higher engagement rates, indicating potential user behaviour patterns.
    2. Email Campaign Effectiveness: Calculated email open rates for different user segments. Found that premium users exhibited a higher open rate, suggesting the effectiveness of targeted premium user campaigns.
    3. Device Specific Engagement: Discovered that mobile devices were the most popular among users, emphasizing the need for a mobile-friendly user interface.
    4. User Segmentation Impact: Identified significant differences in engagement between regular and premium users. Tailored email campaigns for premium users resulted in higher engagement rates compared to generic campaigns.

Result

- ✓ Through the project, a profound understanding of user behaviour and operational trends was achieved.
- ✓ Insights into user activity patterns, language preferences, and event throughputs provided valuable directions for business decisions.
- ✓ The identification and resolution of duplicate data enhanced data quality, ensuring more accurate analysis.
- ✓ The outcomes contributed significantly to informed decision-making, leading to optimized operations and enhanced user experience.
- ✓ By understanding user behaviour, email campaigns were optimized, leading to higher open rates and click-through rates.
- ✓ Identified the importance of mobile devices, leading to UI/UX enhancements for mobile users, improving overall user experience.
- ✓ The project provided actionable insights, enabling data-driven decisions. Tailoring strategies based on user behaviour resulted in more effective marketing campaigns and improved user engagement, contributing to the company’s growth and customer satisfaction.

CASE STUDY 1: Job Data Analysis

Task-1

```sql
# Write an SQL query to calculate the no. of jobs reviewed per hour for each day in November 2020.
SELECT DATE_FORMAT(STR_TO_DATE(ds, '%m/%d/%Y'), '%m/%d/%Y') AS review_date,
       COUNT(job_id) AS jobs_reviewed_per_hour
FROM job_data
WHERE EXTRACT(MONTH FROM STR_TO_DATE(ds, '%m/%d/%Y')) = 11
  AND EXTRACT(YEAR FROM STR_TO_DATE(ds, '%m/%d/%Y')) = 2020
GROUP BY review_date
ORDER BY review_date;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

review_date	jobs_reviewed_per_hour
11/25/2020	1
11/26/2020	1
11/27/2020	1
11/28/2020	2
11/29/2020	1
11/30/2020	2

The output will provide a list of dates and the number of jobs reviewed in each hour. This information can be valuable for identifying peak review times during November 2020.

This insight can be used to optimize resource allocation, schedule tasks, or investigate further to understand the reasons behind the spikes in job reviews during specific hours on particular days.

Task-2

```sql
# Calculate the 7-day rolling average of throughput.
SELECT a.ds,
       AVG(b.events_count) AS rolling_avg
FROM (SELECT ds, COUNT(*) AS events_count FROM job_data GROUP BY ds) AS a
JOIN (SELECT ds, COUNT(*) AS events_count FROM job_data GROUP BY ds) AS b
ON STR_TO_DATE(b.ds, '%m/%d/%Y')
BETWEEN DATE_SUB(STR_TO_DATE(a.ds, '%m/%d/%Y'), INTERVAL 6 DAY) AND STR_TO_DATE(a.ds, '%m/%d/%Y')
GROUP BY a.ds
ORDER BY a.ds;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	rolling_avg
11/25/2020	1.0000
11/26/2020	1.0000
11/27/2020	1.0000
11/28/2020	1.2500
11/29/2020	1.2000
11/30/2020	1.3333

Using both daily metrics and rolling average in conjunction can provide a comprehensive view.

Daily metrics can help in investigating short-term changes.
Rolling average can provide context by highlighting overarching trends.

Task-3

```sql
# Write an SQL query to calculate the percentage share of each language over the last 30 days.
SELECT language,
       COUNT(*) * 100.0 / (SELECT COUNT(*) FROM job_data) AS percentage_share
FROM job_data
GROUP BY language;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

language	percentage_share
English	12.50000
Arabic	12.50000
Persian	37.50000
Hindi	12.50000
French	12.50000
Italian	12.50000

The output of this query will provide a breakdown of the percentage share of each language in the jobs reviewed over the last 30 days.

This information can offer valuable insights into the language preferences of the users or the distribution of content in different languages on the platform.

By regularly running this query and monitoring the language distribution, you can make data-driven decisions to enhance user experience, optimize content delivery, and improve customer satisfaction based on the language preferences of your audience.

Task-4

```sql
# Write an SQL query to display duplicate rows from the job_data table.
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);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	job_id	actor_id	event	language	time_spent	org

Handling duplicates is a common data cleaning task in any database management process. Regularly running such checks ensures the database remains accurate and reliable, supporting informed decision-making and analysis within the organization.

CASE STUDY 2: Investigating Metric Spike

Task-1

```
#Write an SQL query to calculate the weekly user engagement.
SELECT
    u.user_id,
    u.company_id,
    WEEK(e.occured_at) AS week_number,
    COUNT(*) AS num_events
FROM
    users u
JOIN
    events e ON u.user_id = e.user_id
GROUP BY
    u.user_id, u.company_id, week_number
ORDER BY
    week_number ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

user_id	company_id	week_number	num_events
10522	1147	17	6
10612	11066	17	12
11212	5535	17	8
11077	2125	17	7
11037	471	17	25
5424	3345	17	21
9874	277	17	4
11702	141	17	6
8869	4505	17	6
11215	486	17	2
11240	2	17	12
11135	407	17	9
10576	3565	17	24
10309	13151	17	13
8807	7	17	10
8269	2454	17	6
11364	3018	17	15

By analyzing weekly user engagement, you can identify patterns and trends.
A consistent or increasing number of events per week might indicate growing user engagement, while a declining trend might signal disengagement or issues with the platform.

Task-2

```sql
# Write an SQL query to calculate the user growth for the product.
SELECT
    EXTRACT(YEAR FROM created_at) AS year,
    EXTRACT(MONTH FROM created_at) AS month,
    COUNT(DISTINCT user_id) AS new_users
FROM
    users
GROUP BY
    year, month
ORDER BY
    year, month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	month	new_users
2013	1	160
2013	2	160
2013	3	150
2013	4	181
2013	5	214
2013	6	213
2013	7	284
2013	8	316
2013	9	330
2013	10	390
2013	11	399
2013	12	486
2014	1	552
2014	2	525
2014	3	615
2014	4	726
2014	5	779
2014	6	873
2014	7	997
2014	8	1031

This query provides a clear view of user growth on a monthly basis. By comparing the number of new users each month, you can track growth trends over time. Identifying patterns in user growth can provide valuable insights.

For example, recurring spikes in certain months might be related to marketing campaigns or seasonal factors. Understanding these patterns can help in planning future marketing strategies.

Task-3

```sql
#Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.
SELECT EXTRACT(YEAR FROM u.created_at) AS signup_year,
       EXTRACT(YEAR FROM e.occured_at) AS activity_year,
       EXTRACT(WEEK FROM e.occured_at) AS activity_week,
       COUNT(DISTINCT u.user_id) AS total_users,
       COUNT(DISTINCT CASE WHEN EXTRACT(WEEK FROM e.occured_at) = EXTRACT(WEEK FROM u.created_at) THEN u.user_id END) AS retained_users,
       (COUNT(DISTINCT CASE WHEN EXTRACT(WEEK FROM e.occured_at) = EXTRACT(WEEK FROM u.created_at) THEN u.user_id END) * 100.0) / COUNT(DISTINCT u.user_id) AS retention_rate
FROM users u
JOIN events e ON u.user_id = e.user_id
WHERE e.occured_at >= u.created_at
GROUP BY signup_year, activity_year, activity_week
ORDER BY signup_year, activity_year, activity_week;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

signup_year	activity_year	activity_week	total_users	retained_users	retention_rate
2013	2014	17	232	3	1.29310
2013	2014	18	380	8	2.10526
2013	2014	19	398	10	2.51256
2013	2014	20	393	3	0.76336
2013	2014	21	362	3	0.82873
2013	2014	22	384	2	0.52083
2013	2014	23	399	2	0.50125
2013	2014	24	419	5	1.19332
2013	2014	25	384	6	1.56250
2013	2014	26	387	4	1.03359
2013	2014	27	379	6	1.58311
2013	2014	28	383	8	2.08877
2013	2014	29	372	10	2.68817
2013	2014	30	383	8	2.08877
2013	2014	31	310	5	1.61290
2013	2014	32	257	6	2.33463
2013	2014	33	245	8	3.26531
2013	2014	34	235	3	1.27660
2013	2014	35	10	1	10.00000

The query calculates the retention rate by comparing the number of users active in the same week they signed up (retained_users) with the total number of users who signed up (active_users). The retention rate indicates what percentage of users remained engaged after their initial interaction with the product.

A steady or increasing retention rate over weeks suggests that users are finding value in the product. Sudden drops in retention might indicate issues with user experience or product satisfaction.

Users' first experiences with a product are crucial. Analyzing retention rates in the initial weeks after sign-up helps evaluate the effectiveness of the on boarding process. A high retention rate indicates that users are finding value early on, while a low rate might suggest a need for improvements in the on boarding process.

Task-4

```sql
#Write an SQL query to calculate the weekly engagement per device.
SELECT
    WEEK(occured_at) AS week_number,
    device,
    COUNT(*) AS num_events
FROM
    events
GROUP BY
    week_number, device
ORDER BY
    week_number, num_events DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

week_number	device	num_events
17	macbook pro	1527
17	lenovo thinkpad	801
17	iphone 5	715
17	dell inspiron notebook	506
17	macbook air	493
17	iphone 5s	476
17	samsung galaxy s4	454
17	nexus 5	385
17	ipad air	331
17	asus chromebook	254
17	iphone 4s	219
17	ipad mini	208
17	acer aspire notebook	207
17	htc one	192
17	dell inspiron desktop	188
17	nexus 7	181
17	nexus 10	145
17	hp pavilion desktop	134
17	nokia lumia 635	130
17	samsung galaxy note	117
17	windows surface	87
17	amazon fire phone	84

The query provides a breakdown of user activity based on devices. This information can help identify which devices are most commonly used for accessing the product and the level of engagement on each device type.

Understanding which devices users are most active on helps in optimizing the product for specific devices. If a particular device type shows consistently high or low activity, it can influence development decisions and user interface design.

Task-5

```sql
# Write an SQL query to calculate the email engagement metrics.
SELECT COUNT(*) AS total_emails_sent,
       SUM(CASE WHEN action = 'email_open' THEN 1 ELSE 0 END) AS total_emails_opened,
       SUM(CASE WHEN action = 'email_clickthrough' THEN 1 ELSE 0 END) AS total_emails_clicked,
       SUM(CASE WHEN action = 'sent_weekly_digest' THEN 1 ELSE 0 END) AS total_digests,
       ROUND(SUM(CASE WHEN action = 'email_open' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) AS open_rate,
       ROUND(SUM(CASE WHEN action = 'email_clickthrough' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) AS click_through_rate,
       ROUND(SUM(CASE WHEN action = 'sent_weekly_digest' THEN 1 ELSE 0 END) * 100.0 / SUM(CASE WHEN action = 'email_clickthrough' THEN 1 ELSE 0 END), 2) AS digest_rate
FROM email_events;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_emails_sent	total_emails_opened	total_emails_clicked	total_digests	open_rate	click_through_rate	digest_rate
90389	20459	9010	57267	22.63	9.97	635.59

Analyzing email actions over time and across user types provides insights into how users engage with email content. Patterns, such as increased engagement during weekends or higher open rates among premium users, can guide email campaign scheduling and targeting strategies.

By tracking user actions, you can identify the types of content (promotions, newsletters, updates) that resonate most with users. Insights gained can inform content creation strategies and help tailor emails to match user preferences.