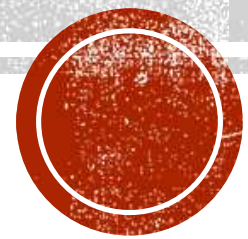
OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE

Presented by :-

Souvik Karmakar



DESCRIPTION:

- Embark on a journey into the realm of Operational Analytics with this engaging project, where the Lead Data Analyst plays a pivotal role in unraveling the intricacies of a company's operations. Collaborating closely with teams in operations, support, and marketing, the Lead Data Analyst tackles daily challenges, particularly in investigating metric spikes—unexpected shifts in critical indicators such as daily user engagement and sales.

- Armed with advanced SQL skills, the Lead Data Analyst utilizes provided datasets and tables to answer questions from different departments, mirroring the responsibilities of a lead data analyst at a prominent company like Microsoft. This project goes beyond mere data analysis; it emphasizes leveraging analytical prowess to provide insights that drive operational enhancements and unveil the reasons behind sudden changes in key metrics. Participants can anticipate making a tangible impact on the company's success through astute, data-driven decision-making.

OBJECTIVE 1: JOB DATA ANALYSIS

Overview: The project involves a comprehensive analysis of job-related data stored in a table named **job_data**. The table comprises essential columns such as **job_id**, **actor_id**, **event**, **language**, **time_spent**, **org**, and **ds** (date in yyyy/mm/dd format). The primary focus is to extract meaningful insights from this dataset, shedding light on job-related activities.

Key Columns:

1. **job_id:** Unique identifier for each job.
2. **actor_id:** Unique identifier for each actor.
3. **event:** Type of event, categorized as decision, skip, or transfer.
4. **language:** Language of the content associated with the job.
5. **time_spent:** Duration spent reviewing the job in seconds.
6. **org:** Organization of the actor undertaking the job.
7. **ds:** Date of the event in yyyy/mm/dd format (stored as text).

OBJECTIVE 2: INVESTIGATING METRIC SPIKE

Overview: This case study involves investigating a metric spike using three tables: **users**, **events**, and **email_events**. Each table serves a distinct purpose in providing information about users, their actions, and specific events related to email activities. The primary goal is to identify and understand the causes behind a sudden spike in a key metric.

Key Tables:

1. **users:** Contains descriptive information about each user's account.
2. **events:** Captures various user actions, such as login, messaging, and search.
3. **email_events:** Focuses specifically on events related to the sending of emails.

APPROACH:-

The project was approached by first understanding the requirements and questions posed by different departments. The relevant datasets, such as jobData, users, events, and emailEvents, were collected and analyzed using SQL queries. The queries were designed to calculate specific metrics and derive meaningful insights. The results were then examined, and patterns and trends were identified. The analysis was performed iteratively, refining the queries and adjusting the approach as needed to ensure accurate and relevant insights.

Tech-Stack Used:-

The project utilized SQL for data analysis and query execution. The SQL queries were executed using tools such as **MySQL Workbench 8.0.35.0 Community Edition**. The purpose of using SQL was to efficiently extract, transform, and analyse the data from the provided datasets. The use of SQL allowed me for easy manipulation of the datasets and enabled the calculation of various metrics and insights based on the specific requirements.

Case Study 1 :– Jobs data:

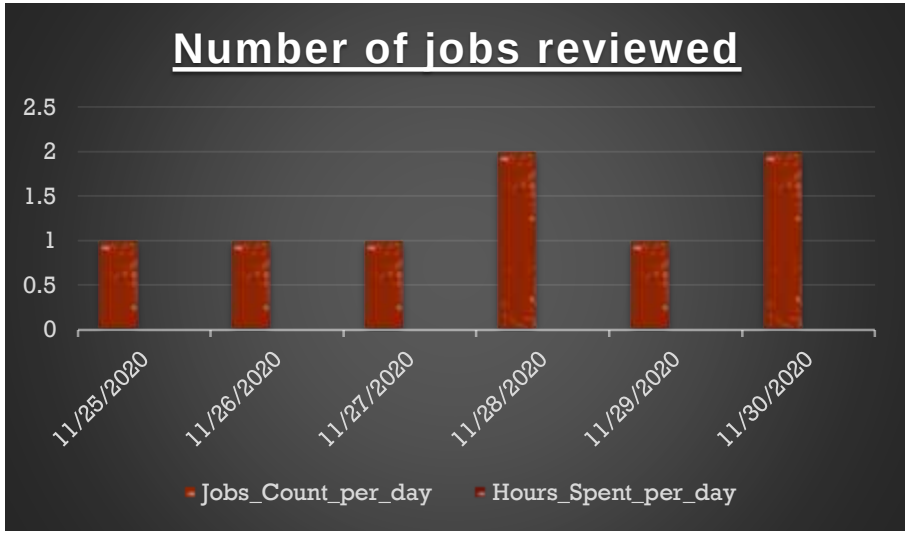
A. **Number of jobs reviewed:** Amount of jobs reviewed over time.

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

SQL QUERY:-

```
SELECT ds AS Date,
COUNT(job_id) AS
Jobs_Count_per_day,
sum(time_spent) / 3600
AS Hours_Spent_per_day
FROM jobData WHERE
ds >= '2020-11-01' AND
ds <= '2020-11-30'
GROUP BY ds;
```

Date	Jobs_Count_per_day	Hours_Spent_per_day
2020-11-30	2	0.0111
2020-11-29	1	0.0056
2020-11-28	2	0.0092
2020-11-27	1	0.0289
2020-11-26	1	0.0156
2020-11-25	1	0.0125



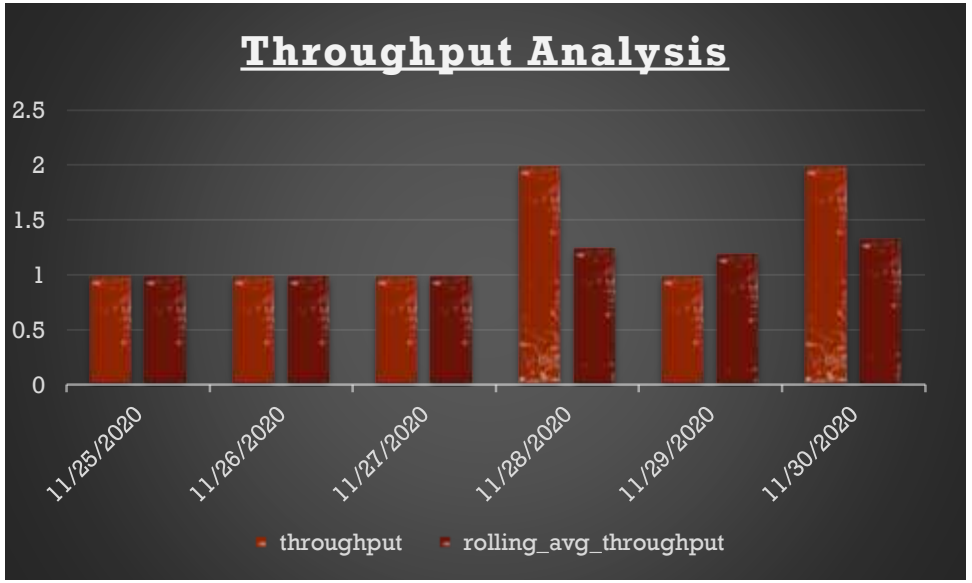
B. Throughput Analysis: Calculate the 7-day rolling average of throughput (number of events per second).

Task: Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

SQL Query:-

```
SELECT ds, COUNT(*) AS
throughput, AVG(COUNT(*))
OVER(ORDER BY ds ROWS
BETWEEN 6 PRECEDING AND
CURRENT ROW) AS
rolling_avg_throughput FROM
jobData GROUP BY ds;
```

ds	throughput	rolling_avg_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



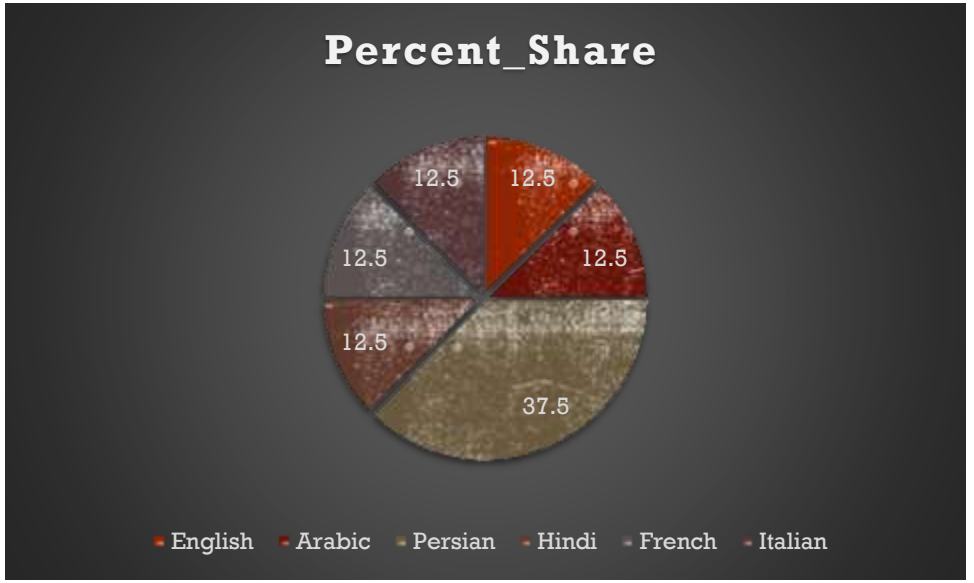
C.<u>Language Share Analysis:</u> Calculate the percentage share of each language in the last 30 days.

<u>Task:</u> Write an SQL query to calculate the percentage share of each language over the last 30 days.

SQL Query:-

```
with CommonT as (SELECT
language, count(job_id) AS
no_of_jobs FROM jobData
GROUP BY language)
SELECT *,
ROUND(no_of_jobs*100/(SEL
ECT sum(no_of_jobs) FROM
CommonT),2) as
Percent_Share FROM
CommonT;
```

language	no_of_jobs	Percent_Share
English	1	12.50
Arabic	1	12.50
Persian	3	37.50
Hindi	1	12.50
French	1	12.50
Italian	1	12.50



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

Task: Write an SQL query to display duplicate rows from the jobData table.

SQL Query:-

```
SELECT ds, job_id,
actor_id, org,
COUNT(*) AS
occurences FROM
jobData GROUP BY ds
, job_id , actor_id , org
HAVING occurences >
1;
```

Result:-

	ds	job_id	actor_id	org	occurences

No duplicate rows .

Case Study 2: Investigating Metric Spike

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

Task: Write an SQL query to calculate the weekly user engagement.

SQL Query:-

```
SELECT YEAR(occurred_at) AS year,
WEEK(occurred_at) AS week, COUNT(*)
AS engagement_count
FROM events
WHERE event_type = 'engagement'
GROUP BY YEAR(occurred_at),
WEEK(occurred_at),YEAR(occurred_at);
```

year	week	user_engagement_count
2014	17	8019
2014	18	17341
2014	19	17224
2014	20	17911
2014	21	17151
2014	23	18280
2014	22	18413
2014	24	19052
2014	25	18642
2014	29	20067
2014	26	19061
2014	30	21533
2014	28	20776
2014	27	19881
2014	31	18556
2014	32	16612
2014	33	16145
2014	34	16127
2014	35	784

Result:-


Chart Title
25000
20000
15000
10000
5000
0
17 8019
18 17341
19 17224
20 17911
21 17151
23 18280
22 18413
24 19052
25 18642
29 20067
26 19061
30 21533
28 20776
27 19881
31 18556
32 16612
33 16145
34 16127
35 784
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
week
user_engagement_count

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

Task: Write an SQL query to calculate the user growth for the product.

SQL Query:-

```
SELECT DATE_FORMAT(created_at,
'%Y-%m') AS month,
COUNT(DISTINCT user_id) AS
user_growth,
ROUND((COUNT(DISTINCT
user_id) - LAG(COUNT(DISTINCT
user_id)) OVER(ORDER BY
DATE_FORMAT(created_at, '%Y-
%m'))) / LAG(COUNT(DISTINCT
user_id))OVER (ORDER BY
DATE_FORMAT(created_at, '%Y-
%m')) * 100, 1) AS growth_rate
FROM users GROUP BY month
ORDER BY month;
```

month	user_growth	growth_rate
2013-01	160	NULL
2013-02	1 160	0.0
2013-03	150	-6.3
2013-04	181	20.7
2013-05	214	18.2
2013-06	213	-0.5
2013-07	284	33.3
2013-08	316	11.3
2013-09	330	4.4
2013-10	390	18.2
2013-11	399	2.3
2013-12	486	21.8
2014-01	552	13.6
2014-02	525	-4.9
2014-03	615	17.1
2014-04	726	18.0
2014-05	779	7.3
2014-06	873	12.1
2014-07	997	14.2
2014-08	1031	3.4

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

Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.

SQL Query:-

```
SELECT EXTRACT(year FROM occured_at) AS Year, EXTRACT(week FROM occured_at) AS Week, device,
COUNT(DISTINCT(user_id)) as Device_no
FROM events WHERE event_type = 'engagement' GROUP BY 1,2,3 ORDER BY 1,2,3 ASC;
```

Year	Week	device	Device_no
2014	17	acer aspire desktop	9
2014	17	acer aspire notebook	20
2014	17	amazon fire phone	4
2014	17	asus chromebook	21
2014	17	dell inspiron desktop	18
2014	17	dell inspiron notebook	46
2014	17	hp pavilion desktop	14
2014	17	htc one	16
2014	17	ipad air	27
2014	17	ipad mini	19
2014	17	iphone 4s	21
2014	17	iphone 5	65
2014	17	iphone 5s	42
2014	17	kindle fire	6
2014	17	lenovo thinkpad	86
2014	17	mac mini	6
2014	17	macbook air	54
2014	17	macbook pro	143
2014	17	nexus 10	16
2014	17	nexus 5	40
2014	17	nexus 7	18
2014	17	nokia lumia 635	17
2014	17	samsumg galaxy tablet	8
2014	17	samsung galaxy note	7
2014	17	samsung galaxy s4	52
2014	17	windows surface	10
2014	18	acer aspire desktop	26
2014	18	acer aspire notebook	33
2014	18	amazon fire phone	9
2014	18	asus chromebook	42
2014	18	dell inspiron desktop	58
2014	18	dell inspiron notebook	77
2014	18	hp pavilion desktop	37
2014	18	htc one	19
2014	18	ipad air	52
2014	18	ipad mini	30
2014	18	iphone 4s	46
2014	18	iphone 5	113
2014	18	iphone 5s	73
2014	18	kindle fire	27
2014	18	mac mini	13
2014	18	macbook air	121
2014	18	macbook pro	252
2014	18	nexus 10	30
2014	18	nexus 5	73
2014	18	nexus 7	30
2014	18	nokia lumia 635	33
2014	18	samsumg galaxy tablet	11
2014	18	samsung galaxy note	15
2014	18	samsung galaxy s4	82
2014	18	windows surface	10
2014	19	acer aspire desktop	23
2014	19	acer aspire notebook	41
2014	19	amazon fire phone	12
2014	19	asus chromebook	27
2014	19	dell inspiron desktop	36
2014	19	dell inspiron notebook	83
2014	19	hp pavilion desktop	40
2014	19	htc one	30
2014	19	ipad air	55

Result:-

Link for the full result:- Retention rate result.xlsx

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

Task: Write an SQL query to calculate the weekly engagement per device.

SQL Query:

```
SELECT
DATE_FORMAT(occurred_at,'%Y-%u') AS week,
device,
COUNT(*) AS event_count,
COUNT(DISTINCT user_id) AS unique_users,
COUNT(*) / COUNT(DISTINCT user_id) AS
engagement_per_user
FROM events
GROUP BY week, device
ORDER BY week, device;
```

week	device	event_count	unique_users	engagement_per_user
2014-18	acer aspire desktop	71	10	7.1000
2014-18	ac	5	21	10.2381
2014-18	amazon fire phone	84	4	21.0000
2014-18	asus chromebook	286	23	12.4348
2014-18	dell inspiron desktop	198	21	9.4286
2014-18	dell inspiron notebook	569	49	11.6122
2014-18	hp pavilion desktop	141	15	9.4000
2014-18	htc one	192	16	12.0000
2014-18	ipad air	358	30	11.9333
2014-18	ipad mini	229	21	10.9048
2014-18	iphone 4s	219	21	10.4286
2014-18	iphone 5	774	70	11.0571
2014-18	iphone 5s	512	45	11.3778
2014-18	kindle fire	64	6	10.6667
2014-18	lenovo thinkpad	858	90	9.5333

Result:-

Link for full result - Weekly engagement rate.xlsx

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

Task: Write an SQL query to calculate the email engagement metrics.

SQL Query:-

```
SELECT ROUND
((100.0 *SUM(CASE WHEN action1 IN
('Email Clicked') THEN 1 ELSE 0
END)/SUM(CASE WHEN action1 IN ('Email
Sent')  THEN 1 ELSE 0 END)),2) AS
Email_Click_Rate, ROUND((100.0 *SUM(CASE
WHEN action_details  IN ('Email Opened')
THEN 1 ELSE 0 END)/SUM(CASE WHEN
action_details IN ('Email Sent')  THEN 1 ELSE 0
END)),2) AS Email_Open_Rate FROM
Email_Data;
```

Email_Click_Rate	Email_Open_Rate
14.79	33.58

CONCLUSION:-

- In conclusion, operational analytics and the investigation of metric spikes are crucial aspects of business management that should be conducted regularly, aligning with the specific needs of the firm. This analysis can be performed on a daily, weekly, monthly, quarterly, or yearly basis, depending on the business requirements and objectives.
- Furthermore, prioritizing email engagement with customers is essential for fostering a strong relationship and expanding the customer base. Crafting compelling email content with attractive headings, coupled with offering reasonable discounts and coupons, can contribute significantly to customer retention and acquisition.
- Additionally, it is advisable for firms to establish a dedicated department, if feasible, to address the concerns of visitors who abandon the sign-up process. Providing guidance and support to these individuals can be instrumental in converting them from mere visitors into valued customers. This proactive approach demonstrates a commitment to customer satisfaction and contributes to the overall success of the business.