DSML AUG24 BEGINNER BATCH

Business Case: Target SQL

Submitted By :- Kashish Bhadauriya

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

1) Data type of all columns in the "customers" table.

```
select column_name, data_type
from `sql_b_case1.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name= 'customers'
```

Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

2) Get the time range between which the orders were placed.

```
select
concat(min(order_purchase_timestamp),' - ',max(order_purchase_timestamp))
as order_placed_range
from `sql_b_case1.orders`
```

Row	order_placed_range
1	2016-09-04 21:15:19+00 - 2018-10-17 17:30:18+00

3) Count the cities and states of the customers who ordered during the given period.

```
select count(distinct c.customer_city) as city_count,
count(distinct c.customer_state) as state_count
from `sql_b_case1.customers` c
join `sql_b_case1.orders` o
on c.customer_id=o.customer_id
where o.order_purchase_timestamp between
(select min(order_purchase_timestamp) from `sql_b_case1.orders`) and
(select max(order_purchase_timestamp) from `sql_b_case1.orders`)
```

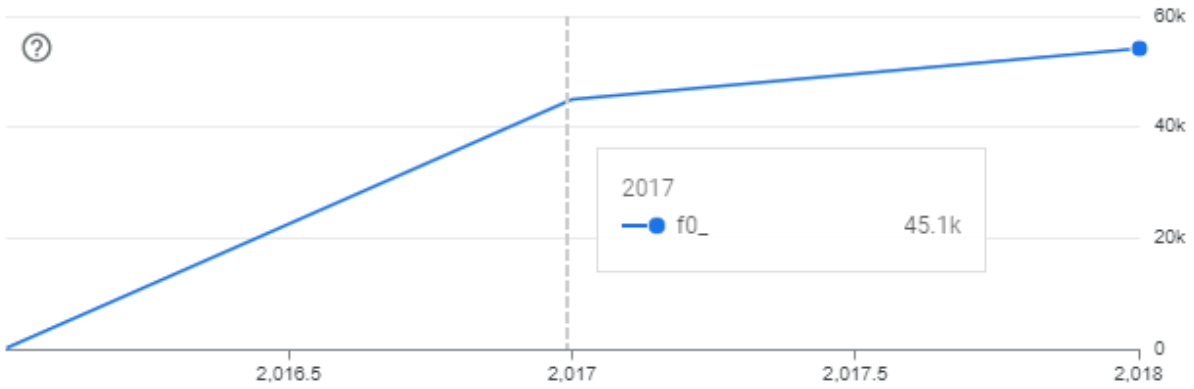
Row	city_count	state_count
1	4119	27

2. In-depth Exploration:

1) Is there a growing trend in the no. of orders placed over the past years?

```
select extract(year from order_purchase_timestamp) as year,
count(order_id) as order_count
from `sql_b_case1.orders`
group by year
order by year
```

Row	year	order_count
1	2016	329
2	2017	45101
3	2018	54011



Insights:- The growth from 2016 to 2017 is substantial, showing an increase of approximately 13,600% in the number of orders, indicating a significant boost in business or customer base.

From 2017 to 2018, there was an additional increase of about 19.8% in the number of orders, showing continuous growth but at a slower rate as compared to the previous rate.

2) Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

```
select extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(order_id) as order_count
from `sql_b_case1.orders`
group by year,month
order by year,month
```

Row	year	month	order_count
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269
17	2018	2	6728
18	2018	3	7211
19	2018	4	6939
20	2018	5	6873
21	2018	6	6167
22	2018	7	6292
23	2018	8	6512
24	2018	9	16
25	2018	10	4

Insights:- The order count steadily increases from January to May in both years. This suggests that the first half of the year experiences growing activity in terms of orders.

3) During what of the day, do the Brazilian customers mostly place their orders? (Dawn, morning, afternoon or night)

0-6 hrs :- Dawn

7-12 hrs :- Mornings

13-18 hrs :- Afternoon

19-23 hrs :- Night

```
select purchase_time,
count(customer_id) as order_count
from
(select customer_id,
case
when extract(hour from order_purchase_timestamp) between 0 and 6 then 'Dawn'
when extract(hour from order_purchase_timestamp) between 7 and 12 then 'Mornings'
when extract(hour from order_purchase_timestamp) between 13 and 18 then 'Afternoon'
when extract(hour from order_purchase_timestamp) between 19 and 23 then 'Night'
end as purchase_time
from `sql_b_case1.orders` ) t
group by purchase_time
order by order_count desc
```

Row	purchase_time	order_count
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

Insights:- Brazilian customers place their orders mostly

In the afternoon.

In Brazil, many people work regular business hours from the morning to early afternoon. During lunch breaks or after finishing work for the day, they may have more time to shop online. The afternoon could provide a relaxed time to browse and make purchases.

3. Evolution of E-commerce orders in the Brazil region:

1) Get the month on month no. of orders placed in each state.

```
select c.customer_state,
extract(year from o.order_purchase_timestamp) as year,
extract(month from o.order_purchase_timestamp) as month,
count(o.order_id) as order_count
from `sql_b_case1.orders` o
join `sql_b_case1.customers` c
on o.customer_id=c.customer_id
group by year, month, c.customer_state
order by c.customer_state, year, month
```

Row	customer_state	year	month	order_count
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5

Results per page: 50 1 – 50 of 565

2) How are the customers distributed across all the states?

```
select customer_state,
count(distinct customer_id) as customer_count
from `sql_b_case1.customers`
group by customer_state
order by customer_count
```

Row	customer_state	customer_count
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485
10	PI	495

11	PB	536
12	MS	715
13	MA	747
14	MT	907
15	PA	975
16	CE	1336
17	PE	1652
18	GO	2020
19	ES	2033
20	DF	2140
21	BA	3380
22	SC	3637
23	PR	5045
24	RS	5466
25	MG	11635
26	RJ	12852
27	SP	41746

Insights:- The states with the highest customer counts are DF (2,140), ES (2,033), and GO (2,020).

The states with the lowest customer counts are RR (46), AP (68), and AC (81).

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

1) Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

You can use the "payment_value" column in the payments table to get the cost of orders.

```
with cost as (select
extract(year from o.order_purchase_timestamp) as years,
sum(p.payment_value) as total_payment
from `sql_b_case1.orders` o
join `sql_b_case1.payments` p
on o.order_id=p.order_id
where extract(year from o.order_purchase_timestamp) in (2017,2018) and
extract(month from o.order_purchase_timestamp) between 1 and 8
group by years
order by years)
select round((y2018.total_payment-y2017.total_payment)*100/y2017.total_payment,4)
as percentage_increment
from (select total_payment from cost where years=2017) as y2017,
(select total_payment from cost where years=2018) as y2018
```

Row	percentage_increment
1	136.9769

2) Calculate the Total & Average value of order price for each state.

```
select c.customer_state,
sum(oi.price) as total_order_price,
avg(oi.price) as avg_order_price
from `sql_b_case1.orders` o
join `sql_b_case1.customers` c
on o.customer_id=c.customer_id
join `sql_b_case1.order_items` oi
on o.order_id=oi.order_id
group by c.customer_state
order by total_order_price desc, avg_order_price desc
```

Row	customer_state	total_order_price	avg_order_price
1	SP	5202955.050001...	109.6536291597...
2	RJ	1824092.669999...	125.1178180945...
3	MG	1585308.029999...	120.7485741488...
4	RS	750304.0200000...	120.3374530874...
5	PR	683083.7600000...	119.0041393728...
6	SC	520553.3400000...	124.6535775862...
7	BA	511349.9900000...	134.6012082126...
8	DF	302603.9399999...	125.7705486284...
9	GO	294591.9499999...	126.2717316759...
10	ES	275037.3099999...	121.9137012411...

3) Calculate the Total & Average value of order freight for each state.

```
select c.customer_state,
sum(ot.freight_value) as total_freight,
avg(ot.freight_value) as avg_freight
from `sql_b_case1.order_items` ot
join `sql_b_case1.orders` o
on ot.order_id=o.order_id
join `sql_b_case1.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
order by total_freight desc, avg_freight desc
```

Row	customer_state	total_freight	avg_freight
1	SP	718723.0699999...	15.14727539041...
2	RJ	305589.3100000...	20.96092393168...
3	MG	270853.4600000...	20.63016680630...
4	RS	135522.7400000...	21.73580433039...
5	PR	117851.6800000...	20.53165156794...
6	BA	100156.6799999...	26.36395893656...
7	SC	89660.26000000...	21.47036877394...
8	PE	59449.65999999...	32.91786267995...
9	GO	53114.97999999...	22.76681525932...
10	DF	50625.49999999...	21.04135494596...
11	ES	49764.59999999...	22.05877659574...
12	CE	48351.58999999...	32.71420162381...
13	PA	38699.30000000...	35.83268518518...
14	MA	31523.77000000...	38.25700242718...
15	MT	29715.43000000...	28.16628436018...
16	PB	25719.73000000...	42.72380398671...
17	PI	21218.2	39.14797047970...
18	MS	19144.03000000...	23.37488400488...
19	RN	18860.09999999...	35.65236294896...
20	AL	15914.58999999...	35.84367117117...
21	SE	14111.46999999...	36.65316883116...
22	TO	11732.67999999...	37.24660317460...
23	RO	11417.38000000...	41.06971223021...
24	AM	5478.890000000...	33.20539393939...
25	AC	3686.750000000...	40.07336956521...
26	AP	2788.500000000...	34.00609756097...
27	RR	2235.190000000...	42.98442307692...

5. **Analysis based on sales, freight and delivery time.**

1) Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
   Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
   Do this in a single query.

   You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

   i. time_to_deliver = order_delivered_customer_date - order_purchase_timestamp

   ii. diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date

```
select
order_id,
extract(day from date(order_delivered_customer_date)-date(order_purchase_timestamp))
as time_to_deliver,
extract(day from date(order_delivered_customer_date)-date(order_estimated_delivery_date))
as diff_estimated_delivery
from `sql_b_case1.orders`
where order_delivered_customer_date is not null and
order_purchase_timestamp is not null and
order_estimated_delivery_date is not null
order by time_to_deliver desc, diff_estimated_delivery desc
```

Row	order_id	time_to_deliver	diff_estimated_delive
1	ca07593549f1816d26a572e06...	210	181
2	1b3190b2dfa9d789e1f14c05b...	208	188
3	440d0d17af552815d15a9e41a...	196	165
4	285ab9426d6982034523a855f...	195	166
5	2fb597c2f772eca01b1f5c561b...	195	155
6	0f4519c5f1c541ddec9f21b3bd...	194	161
7	47b40429ed8cce3aee9199792...	191	175
8	2fe324febf907e3ea3f2aa9650...	190	167
9	c27815f7e3dd0b926b5855262...	188	162
10	2d7561026d542c8dbd8f0daea...	188	159

Results per page: 50 1 – 50 of 96476

Insights:- The orders listed show long delivery times, which could indicate potential logistical or regional challenges for these specific orders.

2) Find out the top 5 states with the highest & lowest average freight value.

```
with state_freight as (
select c.customer_state,
round(avg(ot.freight_value),4) as avg_freight
from `sql_b_case1.order_items` ot
join `sql_b_case1.orders` o
on ot.order_id=o.order_id
join `sql_b_case1.customers` c
on o.customer_id=c.customer_id
group by c.customer_state),
state_rank as (
  select customer_state,
  avg_freight ,
  rank() over(order by avg_freight desc) as high,
  rank() over(order by avg_freight asc) as low
  from state_freight
)
```

```sql
select customer_state,
avg_freight,
'Highest' as category
from state_rank
where high<=5
union all
select customer_state,
avg_freight,
'Lowest' as category
from state_rank
where low<=5
order by avg_freight desc
```

Row	customer_state	avg_freight	category
1	RR	42.9844	Highest
2	PB	42.7238	Highest
3	RO	41.0697	Highest
4	AC	40.0734	Highest
5	PI	39.148	Highest
6	DF	21.0414	Lowest
7	RJ	20.9609	Lowest
8	MG	20.6302	Lowest
9	PR	20.5317	Lowest
10	SP	15.1473	Lowest

Insights:- States with higher average freight might face transportation or logistical challenges, either due to geographic isolation or inefficient transportation networks.

States like SP and DF, with significantly lower freight costs, could benefit from better infrastructure or proximity to distribution centers.

3) Find out the top 5 states with the highest & lowest average delivery time.

```sql
with state_delivery_time as (
select c.customer_state,
round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),2)
as avg_delivery_time
from `sql_b_case1.orders` o
join `sql_b_case1.customers` c
on o.customer_id=c.customer_id
group by c.customer_state ),
state_rank as (
  select customer_state,
  avg_delivery_time ,
  rank() over(order by avg_delivery_time desc) as high,
  rank() over(order by avg_delivery_time asc) as low
  from state_delivery_time
)
select customer_state,
avg_delivery_time,
'Highest' as category
from state_rank
where high<=5
union all
select customer_state,
avg_delivery_time,
'Lowest' as category
from state_rank
where low<=5
order by avg_delivery_time desc
```

Row	customer_state	avg_delivery_time	category
1	RR	28.98	Highest
2	AP	26.73	Highest
3	AM	25.99	Highest
4	AL	24.04	Highest
5	PA	23.32	Highest
6	SC	14.48	Lowest
7	DF	12.51	Lowest
8	MG	11.54	Lowest
9	PR	11.53	Lowest
10	SP	8.3	Lowest

Insights:- Northern and northeastern regions of Brazil, such as Roraima, Amapa, Amazonas, Alagoas and Para are facing longer delivery times compared to the south and southeast regions which benefit from better

infrastructure and logistics networks. This indicates a clear correlation between geographic location and delivery efficiency.

4) Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
   You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```
with delivery_speed as (
select c.customer_state,o.order_status,
round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)),2)
as avg_delivery_speed
from `sql_b_case1.orders` o
join `sql_b_case1.customers` c
on o.customer_id=c.customer_id
group by c.customer_state,order_status
)
select customer_state, avg_delivery_speed
from delivery_speed
where avg_delivery_speed>0 and
order_status=lower('delivered')
order by avg_delivery_speed desc
limit 5
```

Row	customer_state	avg_delivery_speed
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

Insights:- The top 5 states where the order delivery is faster than than the estimated date of delivery (Acre, Rondonia, Amapa, Amazonas and

Roraima) are located in the northern region of Brazil. This suggests that geographic location plays a significant role in delivery delays.

6.Analysis based on the payments:

1) Find the month on month no. of orders placed using different payment types.

```
select
extract(year from o.order_purchase_timestamp) as year,
extract(month from o.order_purchase_timestamp) as month,
p.payment_type,
count(o.order_id) as order_count
from `sql_b_case1.orders` o
join `sql_b_case1.payments` p
on o.order_id=p.order_id
group by year,month,payment_type
order by year,month,payment_type
```

Row	year	month	payment_type	order_count
1	2016	9	credit_card	3
2	2016	10	UPI	63
3	2016	10	credit_card	254
4	2016	10	debit_card	2
5	2016	10	voucher	23
6	2016	12	credit_card	1
7	2017	1	UPI	197
8	2017	1	credit_card	583
9	2017	1	debit_card	9
10	2017	1	voucher	61

Results per page: 50 1 – 50 of 90

Insights:-

. Credit cards are consistently one of the most commonly used payment methods across different months and years. It indicates that many customers prefer the flexibility and potential benefits offered by credit cards.

. There is also growing use of UPI that shows that digital payments are becoming more popular as customer move away from the traditional payment methods.

. Debit cards are less popular.

2) Find the no. of orders placed on the basis of the payment installments that have been paid.

```
select p.payment_installments,
count(o.order_id) as order_count
from `sql_b_case1.orders` o
join `sql_b_case1.payments` p
on o.order_id=p.order_id
where p.payment_installments>0
group by payment_installments
order by payment_installments
```

Row	payment_installment	order_count
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328

11	11	23
12	12	133
13	13	16
14	14	15
15	15	74
16	16	5
17	17	8
18	18	27
19	20	17
20	21	3
21	22	1
22	23	1
23	24	18

Insights:- The majority of customers opted to pay in 1 installment, indicating that the large portion of customer base prefers to make full payments rather than spreading the cost over multiple months. This suggests that many customers have higher disposable income or prefer to avoid installment related interest rates.

It also indicates that short-term installment plans are popular.

Recommendation:-

Based on the insights from the data, several key recommendations can be made. Firstly, since the majority of Brazilian customers tend to place orders in the afternoon, optimizing marketing efforts and promotions for these peak hours could lead to higher conversions. Additionally, there is a clear upward trend in order

volume from 2016 to 2018, indicating increasing customer engagement with the platform. It would be beneficial to maintain this momentum by enhancing customer experience, such as through faster delivery services, particularly in states with longer delivery times like RR and AP. For regions with faster-than-expected delivery (like SP), the logistics model could serve as a benchmark for other areas. Finally, offering more flexible installment payment options could attract more customers, as a significant portion of orders were placed with 1 to 3 installment plans. By focusing on these areas, the business can continue to grow and improve customer satisfaction.