TARGET CASE STUDY

BY- UDAY NAGA VAMSI

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

A. Data type of all columns in the "customers" table.

QUERY

```
SELECT
  column_name,
  data_type
FROM
  target_data.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

B. Get the time range between which the orders were placed.

QUERY

```
select
  min(order_purchase_timestamp) as first_order_date,
  max(order_purchase_timestamp) as last_order_date
from
  `target-395006.target_data.orders`;
```

Row	first_order_date	last_order_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

C. Count the number of Cities and States in our dataset.

QUERY

```
select
  count(distinct c.customer_city) as unique_cities,
  count(distinct c.customer_state) as unique_states
from
  `target-395006.target_data.customers` c
left join
  `target-395006.target_data.orders` o
on
  c.customer_id = o.customer_id
```

Row	unique_cities	unique_states
1	4119	27

INSIGHTS:

- Reviewing the data types of columns helps ensure accurate analysis and manipulation. The dataset contains relevant customer information, such as zip codes, cities, and states.
- Understanding the time range of orders enables tracking of business growth over the years. The dataset covers orders placed between **September- 2016** and **October-2018**.
- Analyzing customer locations allows us to identify regions with high order activity, which can guide marketing and distribution strategies. In the given dataset we can observe **4119** unique cities from **27** unique states all together.

II. In-depth Exploration

A. Is there a growing trend in the no. of orders placed in each month over the past years?

QUERY

```
with Year_Month_OrderCount as (
   select extract(year from o.order_purchase_timestamp) as order_year,
        extract(month from o.order_purchase_timestamp) as order_month,
        count(*) as order_count
   from `target-395006.target_data.orders` as o
   group by order_year, order_month
)

select order_year,
     order_month,
     order_count,
     lag(order_count) over ( partition by order_month order by order_year) as prev_month_order_count,
     case
     when order_count - lag(order_count) over ( partition by order_month order by order_year) > 0 then 'INCREASE'
     when order_count - lag(order_count) over ( partition by order_month order by order_year) < 0 then 'DECREASE'
     else "-"
     end
     as Indication
from Year_Month_OrderCount
order by order_month
```

Row	order_year	order_month	order_count	prev_month_order_cc	Indication
1	2017	1	800	*null*	-
2	2018	1	7269	800	INCREASE
3	2017	2	1780	*null*	-
4	2018	2	6728	1780	INCREASE
5	2017	3	2682	*null*	-
6	2018	3	7211	2682	INCREASE
7	2017	4	2404	*null*	-
8	2018	4	6939	2404	INCREASE
9	2017	5	3700	*null*	-
10	2018	5	6873	3700	INCREASE

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

QUERY

```
with Year_Month_OrderCount as (
   select extract(year from o.order_purchase_timestamp) as order_year,
        extract(month from o.order_purchase_timestamp) as order_month,
        count(*) as order_count
   from `target-395006.target_data.orders` as o
   group by order_year, order_month
)
select order_year,
     order_month,
     order_count,
     case
        when order_count > lag(order_count) over (order by order_year, order_month) and
```

```
        order_count > lead(order_count) over (order by order_year, order_month) then 'PEAK'
      else 'NORMAL'
    end as month_type
from Year_Month_OrderCount
order by order_year, order_month;
```

Row	order_year	order_month	order_count	month_type
1	2016	9	4	NORMAL
2	2016	10	324	PEAK
3	2016	12	1	NORMAL
4	2017	1	800	NORMAL
5	2017	2	1780	NORMAL
6	2017	3	2682	PEAK
7	2017	4	2404	NORMAL
8	2017	5	3700	PEAK
9	2017	6	3245	NORMAL
10	2017	7	4026	NORMAL

C. **During what time 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

QUERY

```
select case
        when extract(hour from o.order_purchase_timestamp) between 0 and 6 then 'Dawn'
        when extract(hour from o.order_purchase_timestamp) between 7 and 12 then 'Morning'
        when extract(hour from o.order_purchase_timestamp) between 13 and 18 then 'Afternoon'
        when extract(hour from o.order_purchase_timestamp) between 19 and 23 then 'Night'
    end as time_of_day,
    count(*) as order_count
from `target-395006.target_data.orders` as o
group by time_of_day
```

Row	time_of_day	order_count
1	Morning	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

INSIGHTS:

- There is a consistent **growth** trend in the number of orders placed in each month over the years **2016** and **2017**, indicating sustained business expansion. But when comes to year **2018** there is **decrease** in no. of orders in **November** and **December** months.
- The data suggests potential monthly seasonality, with certain months experiencing higher order volumes like in **Oct** in **2016; Mar, May, Aug, Nov** in **2017** and **Jan, Mar, Aug** in **2018**. Target can plan promotions and resources accordingly.
- Brazilian customers tend to place orders mostly during the **afternoon** hours. Target can optimize staffing and support during peak hours.

III. Evolution of E-commerce orders in the Brazil region

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

QUERY

```
with MonthlyOrderCounts as (
   select extract(year from o.order_purchase_timestamp ) as order_year,
         extract(month from o.order_purchase_timestamp) as order_month,
         c.customer_state,
         count(*) as order_count
   from `target-395006.target_data.orders` as o
   inner join `target-395006.target_data.customers` as c on o.customer_id = c.customer_id
   group by order_year, order_month, c.customer_state
   order by order_year, order_month, c.customer_state
)
select order_year,
      order_month,
      customer_state,
      order_count
from MonthlyOrderCounts
order by order_year, order_month, customer_state;
```

Row	order_year	order_month	customer_state	order_count
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	AL	2
5	2016	10	BA	4
6	2016	10	CE	8
7	2016	10	DF	6
8	2016	10	ES	4
9	2016	10	GO	9
10	2016	10	MA	4

B. How are the customers distributed across all the states?

QUERY

```
select customer_state, count(distinct customer_unique_id) as unique_customers
from `target-395006.target_data.customers`
group by customer_state
order by unique_customers desc
```

Row	customer_state	unique_customers
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952

INSIGHTS:

- Month-on-month order variations across states highlight regional demand patterns. **Customer State – SP** observes most number of orders every year. Target can tailor marketing efforts and inventory management.
- Analyzing customer distribution informs resource allocation for better coverage and delivery. As we can observe from the results **SP, RJ, MG states** holds **most** and **AC, AP, RR** holds **least** number of customers.

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

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

QUERY

```
with orderPayments as (
   select o.order_id,
        extract(year from o.order_purchase_timestamp) as order_year,
        extract(month from o.order_purchase_timestamp) as order_month,
        p.payment_value
   from `target-395006.target_data.orders` as o
   inner join `target-395006.target_data.payments` as 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
)
select order_year,
     order_month,
```

```
    round((sum(payment_value) - lag(sum(payment_value)) over (order by order_year, order_month)) /
        lag(sum(payment_value)) over (order by order_year, order_month) * 100, 2) as percent_increase
from orderPayments
group by order_year, order_month
order by order_year, order_month;
```

Row	order_year	order_month	percent_increase
1	2017	1	*null*
2	2017	2	110.78
3	2017	3	54.11
4	2017	4	-7.13
5	2017	5	41.92
6	2017	6	-13.77
7	2017	7	15.86
8	2017	8	13.84
9	2018	1	65.33
10	2018	2	-10.99

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

QUERY

```
select c.customer_state,
    round(sum(oi.price),2) as total_order_price,
    round(avg(oi.price),2) as average_order_price
from `target-395006.target_data.order_items` as oi
inner join `target-395006.target_data.orders` as o on oi.order_id = o.order_id
```

```
inner join `target-395006.target_data.customers` as c on o.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state;
```

Row	customer_state	total_order_price	average_order_price
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2

Least 5 states with total order price and avg order price

Row	customer_state	total_order_price	average_order_price
1	RR	7829.43	150.57
2	AP	13474.3	164.32
3	AC	15982.95	173.73
4	AM	22356.84	135.5
5	RO	46140.64	165.97

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

QUERY

```
select c.customer_state,
     round(sum(oi.freight_value),2) as total_freight,
     round(avg(oi.freight_value),2) as average_freight
from `target-395006.target_data.order_items` as oi
inner join `target-395006.target_data.orders` as o on oi.order_id = o.order_id
inner join `target-395006.target_data.customers` as c on o.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state;
```

Row	customer_state	total_freight	average_freight
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77
10	MA	31523.77	38.26

INSIGHTS:

- Analyzing cost increase reveals financial trends, which can inform pricing strategies and cost optimization efforts. From the results the cost of orders shows **downward** trend by the month of **Aug** in both 2017 & 2018.
- From the results **RR, AP, AC, AM, RO** states have lowest total order price and average order price as well. Understanding price distribution across states can guide pricing adjustments and promotions.
- Similarly, Freight cost distribution offers insights into shipping efficiency and potential cost-saving measures.

V. Analysis based on sales, freight and delivery time.

A. 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.

QUERY

```
select o.order_id,
    timestamp_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,day) as delivery_time_in_days,
    timestamp_diff(o.order_estimated_delivery_date, o.order_delivered_customer_date, day) as
diff_estimated_delivery_in_days
from `target-395006.target_data.orders` as o;
```

Row	order_id	delivery_time_in_days	diff_estimated_delivery_in_days
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5

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

QUERY

```
with StateFreightAvg as (
   select c.customer_state,
        avg(oi.freight_value) as avg_freight
   from `target-395006.target_data.order_items` as oi
   inner join `target-395006.target_data.orders` as o on oi.order_id = o.order_id
   inner join `target-395006.target_data.customers` as c on o.customer_id = c.customer_id
   group by c.customer_state
)
select customer_state,
      avg_freight,
      case
         when rank() over (order by avg_freight) <= 5 then 'Bottom 5'
```

```
        when rank() over (order by avg_freight desc) <= 5 then 'Top 5'
        else 'Others'
    end as freight_rank
from StateFreightAvg
order by avg_freight;
```

Row	customer_state	avg_freight	freight_rank
1	SP	15.14727539041...	Bottom 5
2	PR	20.53165156794...	Bottom 5
3	MG	20.63016680630...	Bottom 5
4	RJ	20.96092393168...	Bottom 5
5	DF	21.04135494596...	Bottom 5
6	SC	21.47036877394...	Others
7	RS	21.73580433039...	Others
8	ES	22.05877659574...	Others
9	GO	22.76681525932...	Others
10	MS	23.37488400488...	Others

23	PI	39.14797047970...	Top 5
24	AC	40.07336956521...	Top 5
25	RO	41.06971223021...	Top 5
26	PB	42.72380398671...	Top 5
27	RR	42.98442307692...	Top 5

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

QUERY

```
with StateDeliveryavg as (
   select c.customer_state,
        avg(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,day)) as avg_delivery_time
   from `target-395006.target_data.orders` as o
   inner join `target-395006.target_data.customers` as c on o.customer_id = c.customer_id
   group by c.customer_state
)
select customer_state,
     avg_delivery_time,
     case
        when rank() over (order by avg_delivery_time ) <= 5 then 'Fastest 5'
        when rank() over (order by avg_delivery_time desc) <= 5 then 'Slowest 5'
        else 'Others'
     end as delivery_speed
from StateDeliveryavg
order by avg_delivery_time;
```

Row	customer_state	avg_delivery_time	delivery_speed
1	SP	8.298061489072...	Fastest 5
2	PR	11.52671135486...	Fastest 5
3	MG	11.54381329810...	Fastest 5
4	DF	12.50913461538...	Fastest 5
5	SC	14.47956019171...	Fastest 5
6	RS	14.81923652694...	Others
7	RJ	14.84918643244...	Others
8	GO	15.15074092999...	Others
9	MS	15.19115549215...	Others
10	ES	15.33182957393...	Others
23	PA	23.31606765327...	Slowest 5
24	AL	24.04030226700...	Slowest 5
25	AM	25.98620689655...	Slowest 5
26	AP	26.73134328358...	Slowest 5
27	RR	28.97560975609...	Slowest 5

D. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

QUERY

```
with StateDeliverySpeed as (
  select c.customer_state,
```

```
        avg(date_diff(o.order_delivered_customer_date, o.order_estimated_delivery_date,day)) as avg_delivery_speed
    from `target-395006.target_data.orders` as o
    inner join `target-395006.target_data.customers` as c on o.customer_id = c.customer_id
    where o.order_status = 'delivered'
    group by c.customer_state
)
select customer_state,
      avg_delivery_speed,
      case
        when rank() over (order by avg_delivery_speed) <= 5 then 'Fastest 5'
        else 'Others'
      end as delivery_speed_category
from StateDeliverySpeed
order by avg_delivery_speed;
```

Row	customer_state	avg_delivery_speed	delivery_speed_category
1	AC	-19.7625000000...	Fastest 5
2	RO	-19.1316872427...	Fastest 5
3	AP	-18.7313432835...	Fastest 5
4	AM	-18.6068965517...	Fastest 5
5	RR	-16.4146341463...	Fastest 5
6	MT	-13.4311512415...	Others
7	PA	-13.1902748414...	Others
8	RS	-12.9818488023...	Others
9	RN	-12.7573839662...	Others
10	PE	-12.4017576898...	Others

INSIGHTS:

- Analyzing delivery times and discrepancies helps identify operational efficiency and potential areas for improvement.
- Identifying states with high and low average freight values can guide distribution and pricing strategies. **SP, PR, MG, RJ, DF** states stands as **bottom 5** and **PI, AC, RO, PB, RR** states stands as **Top 5** average freight values
- Comparing delivery times across states highlights areas for enhancing logistical efficiency. **SP, PR, MG, DF, SC** states stands as **fastest 5** and **PA, AL, AM, AP, RR** stands as **slowest 5** in delivery times
- Recognizing states with fast delivery relative to estimated dates showcases operational excellence. **AC, RO, AP, AM, RR** states have really fast order delivery

VI. Analysis based on the payments

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

QUERY

```
select extract(YEAR from o.order_purchase_timestamp) as order_year,
    extract(month from o.order_purchase_timestamp) as order_month,
    p.payment_type,
    count(*) as order_count
from `target-395006.target_data.orders` as o
inner join `target-395006.target_data.payments` as p on o.order_id = p.order_id
group by order_year, order_month, p.payment_type
```

```
order by order_year, order_month
```

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

B. Find the no. of orders placed on the basis of the payment installments that havebeen paid.

QUERY

```
Select
    payment_installments,
    count( order_id) as num_orders
from `target-395006.target_data.payments`
where payment_installments > 0
group by payment_installments
order by payment_installments
```

Row	payment_installments	num_orders
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

INSIGHTS:

- Monitoring payment preferences over time can inform marketing strategies and payment platform enhancements. From the results we can observe Most of the payments are done using **Credit Card** as mode of payment.
- Understanding installment payment usage helps tailor payment options and marketing communications.

Actionable Recommendations

- **Promotion Planning**: Capitalize on monthly seasonality by designing promotions and marketing campaigns around peak order months.

- **Logistics Optimization**: Focusing on states with fast delivery times and replicate successful practices in other regions to improve overall delivery efficiency.
- **Pricing Strategies**: Analyzing cost increase trends and consider implementing dynamic pricing models to balance profitability and customer affordability.
- **Payment Convenience**: Offering targeted promotions or discounts for orders with specific payment installments to incentivize usage and enhance customer satisfaction.
- **Customer Outreach**: Leveraging insights from the customer distribution analysis to tailor marketing efforts and build a strong customer base in underserved areas.
- **Operational Efficiency**: Identifying states with high freight costs and long delivery times to address logistical challenges and improve customer experience.

By incorporating these recommendations, Target can optimize its operations, enhance customer satisfaction, and further establish itself as a preferred e-commerce destination in Brazil.