

Business Case Study ---- Target

1.Exploratory analysis:

1.1. Data type of all columns in the “customers” table

```
SELECT
    column_name,
    data_type
FROM
    angelic-gift-399916.target.INFORMATION_SCHEMA.COLUMNS
WHERE
    table_name = 'customers'
;
```

Query results

Press Alt+F1 for accessibility options.

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS CHART PREVIEW EXECUTION GRAPH

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

PERSONAL HISTORY PROJECT HISTORY

REFRESH

Windows taskbar: Search, taskbar icons, system tray (ENG IN, 11:45 PM, 23-09-2023)

Insights:

The `customers` table has 5 columns
`customer_id`, `customer_unique_id`, `customer_city`, `customer_state`, are of
`STRING` datatype and `customer_zip_code_prefix` is of INT64 datatype.

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

```
SELECT
  MIN(order_purchase_timestamp) AS start_range,
  MAX(order_purchase_timestamp) AS end_range,
FROM `target.orders`
;
```

Query results [SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	start_range	end_range					
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC					

PERSONAL HISTORY PROJECT HISTORY [REFRESH](#)

Insights:

All the orders in the dataset were placed during the time period 4th September 2016 to 17th October 2018.

1.3. Count the cities and states of 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 `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
;
```

Query results [SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION RESULTS JSON EXECUTION DETAILS CHART PREVIEW EXECUTION GRAPH

Row	city_count	state_count
1	4119	27

PERSONAL HISTORY PROJECT HISTORY [REFRESH](#)

Windows Search ENG IN 08:18 PM 29-09-2023

Insights:

The orders are distributed among a total of 4119 cities within 27 states.

2. In-depth Exploration:

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

```
WITH orc AS(
  SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) year,
    EXTRACT(MONTH FROM order_purchase_timestamp) month,
    COUNT(*) order_count
  FROM `target.orders`
  GROUP BY year, month
  ORDER BY year, month
)
SELECT
  year,
  month,
  order_count,
  LAG(order_count,1) OVER(ORDER BY year, month) AS prev_month_or,
  order_count - LAG(order_count,1) OVER(ORDER BY year, month) AS diff
FROM orc
ORDER BY year,month;
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		CHART	PREVIEW	EXECUTION GRAPH
Row	year	month	order_count	prev_month_or	diff			
1	2016	9	4	null	null			
2	2016	10	324	4	320			
3	2016	12	1	324	-323			
4	2017	1	800	1	799			
5	2017	2	1780	800	980			
6	2017	3	2682	1780	902			
7	2017	4	2404	2682	-278			
8	2017	5	3700	2404	1296			
9	2017	6	3245	3700	-455			
10	2017	7	4026	3245	781			
11	2017	8	4331	4026	305			
12	2017	9	4285	4331	-46			
13	2017	10	4631	4285	346			

PERSONAL HISTORY

PROJECT HISTORY

Insights:

There is a growing trend in the no. of orders. But the trend has some fluctuations. The order count was at the lowest on Dec 2016 with only 1 order. On Nov 2017 the order count reached its peak with 7544 orders.

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

```
SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
  COUNT(*) AS order_count
FROM `target.orders`
GROUP BY month
ORDER BY month
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	month ▼	order_count ▼					
1	1	8069					
2	2	8508					
3	3	9893					
4	4	9343					
5	5	10573					
6	6	9412					
7	7	10318					
8	8	10843					
9	9	4305					
10	10	4959					
11	11	7544					
12	12	5674					

Insights:

We can observe some seasonality trends in the given dataset. The count of orders generally shows a steady increase till August. There is a sudden sharp fall in the shopping trend in September. After which there is an upward trend for the remainder of the year.

2.3. During what time of the day, do the Brazilian customers mostly place their orders?

```
SELECT
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 "Morning"
  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 ordered_during,
COUNT(*) AS order_count
FROM `target.orders`
GROUP BY ordered_during
ORDER BY order_count DESC
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	ordered_during	order_count					
1	Afternoon	38135					
2	Night	28331					
3	Morning	27733					
4	Dawn	5242					

Insights:

Brazilian customers place most of their orders in the Afternoon. The least number of orders are placed in Dawn.

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

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

```
SELECT
  c.customer_state,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  COUNT(*) AS order_count
FROM `target.orders` o
JOIN `target.customers` c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state, month
ORDER BY c.customer_state, month
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state	month	order_count				
1	AC	1	8				
2	AC	2	6				
3	AC	3	4				
4	AC	4	9				
5	AC	5	10				
6	AC	6	7				
7	AC	7	9				
8	AC	8	7				
9	AC	9	5				
10	AC	10	6				
11	AC	11	5				
12	AC	12	5				
13	AL	1	39				
14	AL	2	39				
15	AL	3	40				
16	AL	4	51				

Insights:

SP has the highest number of orders in any given month. Closely followed by RJ and MG.

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

```
SELECT
  customer_state,
  COUNT(customer_id) AS number_of_customers
FROM `target.customers`
GROUP BY customer_state
ORDER BY number_of_customers DESC
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state ▼	number_of_customers					
1	SP	41746					
2	RJ	12852					
3	MG	11635					
4	RS	5466					
5	PR	5045					
6	SC	3637					
7	BA	3380					
8	DF	2140					
9	ES	2033					
10	GO	2020					
11	PE	1652					
12	CE	1336					
13	PA	975					
14	MT	907					
15	MA	747					
16	MS	715					

Insights:

The state of SP has the highest number of customers. With RJ having the second largest customer base. While MG is the state with the highest number of customers.

4. Impact on Economy:

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

```
SELECT
  tt.month,
  SUM(CASE
    WHEN tt.year = 2017
    THEN tt.cost ELSE 0 END) AS Year2017,
  SUM(CASE
    WHEN tt.year = 2018
    THEN tt.cost ELSE 0 END) AS Year2018,
  (
    (SUM(CASE
      WHEN tt.year = 2018
      THEN tt.cost ELSE 0 END)
    -
    SUM(CASE
      WHEN tt.year = 2017
      THEN tt.cost ELSE 0 END)
    )
    /
    SUM(CASE
      WHEN tt.year = 2017
      THEN tt.cost ELSE 0 END)
  ) * 100 AS percent_increase
FROM (
  SELECT
    EXTRACT(year FROM o.order_purchase_timestamp) year,
    EXTRACT(month FROM o.order_purchase_timestamp) month,
    SUM(p.payment_value) cost
  FROM `target.payments` p
  JOIN `target.orders` o
  ON p.order_id = o.order_id
  WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2017 AND
2018
    AND EXTRACT(month FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
  GROUP BY year, month
  ORDER BY year, month
) tt
GROUP BY month
ORDER BY month
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	month ▼	Year2017 ▼	Year2018 ▼	percent_increase ▼			
1	1	138488.0399999...	1115004.180000...	705.1266954171...			
2	2	291908.0099999...	992463.3400000...	239.9918145445...			
3	3	449863.6000000...	1159652.119999...	157.7786066709...			
4	4	417788.0300000...	1160785.479999...	177.8407701149...			
5	5	592918.8200000...	1153982.149999...	94.62734375677...			
6	6	511276.3800000...	1023880.499999...	100.2596912456...			
7	7	592382.9200000...	1066540.750000...	80.04245463390...			
8	8	674396.3200000...	1022425.320000...	51.60600520477...			

Insights:

We see a 705.12% increase between Jan 2017 and Jan 2018. This is the highest percentage of increase for the data under consideration. The month of Aug shows the least percentage of increase which is only about 51.60 percent.

4.2. Calculate the total and average value of order price for each state

```
SELECT
  c.customer_state,
  ROUND(AVG(oi.price), 2) average_price,
  ROUND(SUM(oi.price), 2) total_price
FROM `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
JOIN `target.order_items` oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY c.customer_state
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state	average_price	total_price				
1	AC	173.73	15982.95				
2	AL	180.89	80314.81				
3	AM	135.5	22356.84				
4	AP	164.32	13474.3				
5	BA	134.6	511349.99				
6	CE	153.76	227254.71				
7	DF	125.77	302603.94				
8	ES	121.91	275037.31				
9	GO	126.27	294591.95				
10	MA	145.2	119648.22				
11	MG	120.75	1585308.03				
12	MS	142.63	116812.64				
13	MT	148.3	156453.53				

Insights:

PB has the highest average price of 191.48. SP has the lowest average price of 109.65.

SP has the highest total price of 52022955.05. RR has the lowest total price which is only about 7829.43.

4.3. Calculate total and average value of order freight for each state

```
SELECT
  c.customer_state,
  ROUND(AVG(oi.freight_value), 2) average_freight_value,
  ROUND(SUM(oi.freight_value), 2) total_freight_value
FROM `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
JOIN `target.order_items` oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY c.customer_state
;
```

← Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state ▼	average_freight_valu	total_freight_value				
1	AC	40.07	3686.75				
2	AL	35.84	15914.59				
3	AM	33.21	5478.89				
4	AP	34.01	2788.5				
5	BA	26.36	100156.68				
6	CE	32.71	48351.59				
7	DF	21.04	50625.5				
8	ES	22.06	49764.6				
9	GO	22.77	53114.98				
10	MA	38.26	31523.77				
11	MG	20.63	270853.46				
12	MS	23.37	19144.03				
13	MT	28.17	29715.43				

Insights:

RR has the highest average freight value 42.98. SP has the lowest average freight value 15.15.

SP has the highest total freight value of 718723.07. RR has the lowest average freight value of 2235.19

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

5.1. 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 and actual delivery date of order. Do this in a single query.

```
SELECT
  order_id,
  date_diff(order_delivered_customer_date,
order_purchase_timestamp, DAY) delivery_time,
  date_diff(order_estimated_delivery_date, order_purchase_timestamp,
DAY) estimated_delivery,
  date_diff(order_estimated_delivery_date,
order_delivered_customer_date, DAY) diff
FROM `target.orders`
WHERE date_diff(order_delivered_customer_date,
order_purchase_timestamp, DAY) IS NOT NULL
ORDER BY order_id
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		CHART	PREVIEW	EXECUTION GRAPH	
Row	order_id	delivery_time	estimated_delivery	diff					
1	00010242fe8c5a6d1ba2dd792...	7	15	8					
2	00018f77f2f0320c557190d7a1...	16	18	2					
3	000229ec398224ef6ca0657da...	7	21	13					
4	00024acbcd0a6daa1e931b03...	6	11	5					
5	00042b26cf59d7ce69dfabb4e...	25	40	15					
6	00048cc3ae777c65dbb7d2a06...	6	21	14					
7	00054e8431b9d7675808bcb8...	8	24	16					
8	000576fe39319847cbb9d288c...	5	20	15					
9	0005a1a1728c9d785b8e2b08...	9	9	0					
10	0005f50442cb953dcd1d21e1f...	2	20	18					
11	00061f2a7bc09da83e415a52d...	4	15	10					
12	00063b381e2406b52ad42947...	10	10	0					
13	0006ec9db01a64e59a68b2c34...	6	28	21					

Insights:

In the given dataset we have orders which whose delivery date exceed the estimated delivery date.

5.2. Find out the top 5 states with the highest and lowest average freight value

```
SELECT
  c.customer_state,
  ROUND(AVG(oi.freight_value), 2) avg_freight_value
FROM `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
JOIN `target.order_items` oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY avg_freight_value DESC
LIMIT 5
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state	avg_freight_value					
1	RR	42.98					
2	PB	42.72					
3	RO	41.07					
4	AC	40.07					
5	PI	39.15					

```
SELECT
  c.customer_state,
  ROUND(AVG(oi.freight_value), 2) avg_freight_value
FROM `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
JOIN `target.order_items` oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY avg_freight_value
LIMIT 5
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state	avg_freight_value					
1	SP	15.15					
2	PR	20.53					
3	MG	20.63					
4	RJ	20.96					
5	DF	21.04					

Insights:

RR has the highest average freight value of all the states in Brazil. The average freight value of RR is 42.98.

SP has the lowest average freight value of all the states in Brazil. The average freight value of SP is 15.15.

5.3. Find out the states with the highest and lowest average delivery time

```
SELECT
  customer_state,
  ROUND(AVG(date_diff(order_delivered_customer_date,
    order_purchase_timestamp, DAY)), 2) average_delivery_time
FROM `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
WHERE date_diff(order_delivered_customer_date, order_purchase_timestamp,
  DAY) IS NOT NULL
GROUP BY customer_state
ORDER BY average_delivery_time DESC
LIMIT 5
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state	average_delivery_time					
1	RR	28.98					
2	AP	26.73					
3	AM	25.99					
4	AL	24.04					
5	PA	23.32					

```
SELECT
  customer_state,
  ROUND(AVG(date_diff(order_delivered_customer_date,
    order_purchase_timestamp, DAY)), 2) average_delivery_time
FROM `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
WHERE date_diff(order_delivered_customer_date, order_purchase_timestamp,
  DAY) IS NOT NULL
GROUP BY customer_state
ORDER BY average_delivery_time
LIMIT 5
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state	average_delivery_time					
1	SP	8.3					
2	PR	11.53					
3	MG	11.54					
4	DF	12.51					
5	SC	14.48					

Insights:

RR has the highest average delivery time. The average delivery time of RR is 28.98 days.

SP has the lowest average delivery time. The average delivery time of SP is 8.3 days.

This finding is in correlation with the previous findings of SP being the state where maximum number of orders are placed.

Thus, we can observe a positive correlation between the no. of orders being placed and the time to deliver them.

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

```
SELECT
  c.customer_state,
  ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY)) , 2)AS avg_diff_in_delivery
FROM `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY avg_diff_in_delivery DESC
LIMIT 5
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customer_state	avg_diff_in_delivery					
1	AC	19.76					
2	RO	19.13					
3	AP	18.73					
4	AM	18.61					
5	RR	16.41					

Insights:

'AC','RO','AP','AM','RR' are the top 5 states where the order delivery is really fast compared to the estimated date of delivery.

6. Analysis based on the payments:

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

```
SELECT
  p.payment_type,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  count(o.order_id) AS order_count
FROM `target.payments` p
JOIN `target.orders` o
ON p.order_id = o.order_id
GROUP BY p.payment_type, month
ORDER BY p.payment_type, month
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	payment_type	month	order_count				
1	UPI	1	1715				
2	UPI	2	1723				
3	UPI	3	1942				
4	UPI	4	1783				
5	UPI	5	2035				
6	UPI	6	1807				
7	UPI	7	2074				
8	UPI	8	2077				
9	UPI	9	903				
10	UPI	10	1056				
11	UPI	11	1509				
12	UPI	12	1160				
13	credit_card	1	6103				
14	credit_card	2	6609				
15	credit_card	3	7707				
16	credit_card	4	7301				

Insights:

Credit cards are the most preferred payment method, followed by UPI. Debit card is the least preferred payment method. There is a high growth in the number of credit card transactions.

There is a up trend in the number of orders placed from Jan to Aug which is in correlation with our previous finding of no. of orders placed.

6.2. 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) order_count
FROM `target.payments` p
JOIN `target.orders` o
ON p.order_id = o.order_id
WHERE order_status != 'canceled' AND payment_installments >=1
GROUP BY p.payment_installments
ORDER BY p.payment_installments
;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	payment_installment	order_count					
1	1	52184					
2	2	12353					
3	3	10392					
4	4	7056					
5	5	5209					
6	6	3898					
7	7	1620					
8	8	4239					
9	9	638					
10	10	5292					
11	11	22					
12	12	133					

Insights:

The least payment installment is 1. The order count of orders with 1 installment is 52184. The installments goes up to 24. There are 18 orders with 24 installments.

Recommendations:

1. On analysing the dataset, we find a positive correlation between the delivery time and the count of orders placed. In the given dataset we find many orders which exceed the estimated delivery time. The orders can be delivered within in the estimated time to improve count of orders. Thus, increasing the company sales.
2. The state of SP has the highest number of orders. There is an area for improving the orders placed in other states. This should be carried out while maintaining the loyalty of customers in SP.
3. We observe seasonal changes in the data. The company can capitalize this peak period for improving the overall sales and expanding the customer base.
4. The data indicates a decline in orders during Sept and Oct. Offering discounts or cash back offers during this period can improve sales during this duration.
5. Most orders are placed using credit cards which is closely followed by UPI. Offering discounts for these payments can be useful for improving sales through the year.