

Analysis on Target Retail's data

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

i) Data type of columns in a table:

The ids are in numeric data type. Order_customer_delivered_date is in datetimestamp. So we need to extract date out of it if in case required.

ii) Time period for which the data is given:

```
Select distinct year from  
(Select EXTRACT (Year FROM CAST(order_purchase_timestamp AS DATE)) as year from DB.orders)
```

Query_Result:

Row	year
1	2017
2	2018
3	2016

Insight: Data of 2016,2017 and 2018 are used

iii) Cities and States covered in the dataset:

```
Select count(distinct customer_state) as state_count,  
count(distinct customer_city) as city_count from `DB.customers`
```

Query_Result:

Row	state_count	city_count
1	27	4119

Insight: i) 27 states and 4119 cities of Brazil are covered

ii) States are given in code.

Recommendation: To get brief about the states, I have used a table code to get the full name of states

2)In-depth Exploration:

i)Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario?
Can we see some seasonality with peaks at specific months?

Select count(distinct order_id) AS Order_Count,Year,Month

FROM

(Select order_id ,

extract(year from cast(order_delivered_customer_date as date)) AS Year,

extract(month from cast(order_delivered_customer_date as date)) AS Month from
'DB.orders'

where order_delivered_customer_date is not null)

group by Year,Month

order by Year desc,count(distinct order_id) desc

Output:

Row	Order_Count	Year	Month
1	8314	2018	8
2	7850	2018	4
3	7111	2018	5
4	6829	2018	6
5	6825	2018	3
6	6597	2018	1
7	5850	2018	2
8	5839	2018	7
9	56	2018	9
10	3	2018	10

Insight: The trend on e-commerce is fluctuating in each year. But there are common months where orders are placed at the peak which are March, July and August but trend has fallen suddenly after huge orders placed.

Recommendation: We can offer discount at the regular interval of time within two months as the purchase count is falling month over month, especially by the end and start of the year.

ii. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

```
Select * FROM (Select distinct hour,(count(distinct order_id) over (partition by hour)) AS order_count,  
CASE
```

```
WHEN hour >= 5 AND hour <6 THEN 'Dawn'
```

```
WHEN hour >= 6 AND hour <12 THEN 'Morning'
```

```
WHEN hour >= 12 AND hour <18 THEN 'Afternoon'
```

```
WHEN (hour >=18 AND hour <=24 OR hour<5 )THEN 'Night'
```

```
END AS status
```

```
FROM (SELECT order_id ,extract(HOUR FROM order_purchase_timestamp) AS hour
```

```
FROM `DB.orders` where order_purchase_timestamp is not null))
```

```
order by order_count desc
```

Query Result

Row	hour	order_count	status
1	16	6675	Afternoon
2	11	6578	Morning
3	14	6569	Afternoon
4	13	6518	Afternoon
5	15	6454	Afternoon
6	21	6217	Night
7	20	6193	Night
8	10	6177	Morning
9	17	6150	Afternoon
10	12	5995	Afternoon

Insight : Brazilian customers prefer Afternoon then Morning to purchase.

Recommendation: Offers or Sale should be offered in Night as customers are placing least so offering in night may attract to purchase.

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

i)Get month on month orders by region, states

```
SELECT Count(order_id) AS  
order_count,order_purchase_timestamp,geolocation_state,geolocation_city,extract(MONTH FROM  
order_purchase_timestamp) AS MONTH from `DB.orders` o
```

```
join `DB.customers` c on o.customer_id=c.customer_id
```

```
join `DB.geolocation` g on g.geolocation_zip_code_prefix=c.customer_zip_code_prefix
```

```
Where order_status = 'created' Group BY  
order_purchase_timestamp,geolocation_state,geolocation_city
```

Query result

Row	order_count	order_purchase_timestamp	state	geolocation_city	MONTH
1	304	2017-11-25 11:10:33 UTC	Rio de Janeiro	rio de janeiro	11
2	73	2017-12-05 01:07:52 UTC	Sao Paulo	general salgado	12
3	68	2017-12-05 01:07:58 UTC	Rio Grande do Sul	sao leopoldo	12
4	10	2017-12-05 01:07:58 UTC	Rio Grande do Sul	são leopoldo	12
5	3	2017-11-06 13:12:34 UTC	Parana	paranavai	11
6	1	2017-11-06 13:12:34 UTC	Parana	paranavai	11

Insight: Maximum order are placed in November in rio de janeiro city. And least order is placed by customers from Parana.

Recommendation: Sale/Discount can be offered in month of November and December which may affect.

ii) How are customers distributed in Brazil

```
Select count(distinct(c.customer_id)) as customer_count,c.customer_city as customer_city,s.state as customer_state,c.customer_zip_code_prefix
from `DB.customers` c
join `DB.geolocation` g on c.customer_zip_code_prefix=g.geolocation_zip_code_prefix
join `DB.orders` o on c.customer_id=o.customer_id
join `DB.scode` s ON g.geolocation_state=s.code
group by customer_city,customer_state,c.customer_zip_code_prefix
Order by count(distinct(c.customer_id)) desc
```

Row	customer_count	customer_city	customer_state	customer_zi...
1	142	rio de janeiro	Rio de Janeiro	22790
2	124	niteroi	Rio de Janeiro	24220
3	121	rio de janeiro	Rio de Janeiro	22793
4	117	niteroi	Rio de Janeiro	24230
5	110	rio de janeiro	Rio de Janeiro	22775
6	101	vila velha	Espirito Santo	29101
7	95	jundiai	Sao Paulo	13212
8	93	ipatinga	Minas Gerais	35162
9	89	rio de janeiro	Rio de Janeiro	22631
10	87	uberlandia	Minas Gerais	38400

Insight: Customers are maximum from rio de janeiro city and least from uberlandia city of Minas Gerais state.

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

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

Create or replace view DB.sales as (Select * FROM

(SELECT o.order_id as order_id,price,order_item_id ,freight_value,shipping_limit_date,

EXTRACT(month FROM shipping_limit_date) AS month,

EXTRACT(Year FROM shipping_limit_date) AS year,

FORMAT_DATE('%B',CAST(shipping_limit_date AS DATE)) AS Month_

FROM `DB.orders` o

JOIN `DB.order_items` ot ON o.order_id=ot.order_id WHERE shipping_limit_date is not null) WHERE year IN (2017,2018) and month <=8);

create or replace view DB.c1 as (SELECT count(*)AS Total_order_2017,Month_,month,SUM(price) AS Total_sale_2017 FROM `DB.sales` WHERE year=2017 group by Month_,month);

create or replace view DB.c2 as (SELECT count(*)AS Total_order_2018,Month_,SUM(price) AS Total_sale_2018 FROM `DB.sales` WHERE year=2018 group by Month_,month);

Select t1.Month_ AS Month,Total_order_2017,Total_order_2018,
((Total_sale_2018-Total_sale_2017)/Total_sale_2017)*100 AS Increase_in_Sale_from_2017_to_2018

from DB.c1 t1 inner join DB.c2 t2 on t1.Month_=t2.Month_ order by t1.month

Row	Month	Total_order_2017	Total_order_2018	Increase_in_Sale_from_2017_to_2018
1	January	681	7492	954.95403791645276
2	February	1866	7375	233.04421327399712
3	March	2751	8759	200.23308295650253
4	April	2364	7637	211.12605124702645
5	May	4150	8765	114.44612701304746
6	June	3801	6897	85.657759188390443
7	July	4116	6672	74.331245496016734
8	August	5042	8815	91.42304030967594

Insight: The sale is increased from 2017 to 2018 for each month.The trend is increasing.

ii).Mean & Sum of price and freight value by customer state

```
CREATE VIEW DB.mean_ as
```

```
(SELECT * FROM
```

```
(SELECT ROUND(SUM(price)/COUNT(price),2) AS Mean_price,
```

```
ROUND (SUM (freight_value)/ COUNT (freight_value),2) AS Mean_freight_value,
```

```
ROUND (SUM (price),2) AS Sum_Price,
```

```
ROUND (SUM (freight_value),2) AS Sum_Freight_Value,
```

```
customer_state
```

```
FROM `DB.order_items` ot
```

```
JOIN `DB.orders` o on ot.order_id =o.order_id
```

```
JOIN `DB.customers` c on c.customer_id=o.customer_id GROUP BY customer_state) ORDER BY  
Mean_price desc,Mean_freight_value desc);
```

```
SELECT Mean_price,Mean_freight_value,Sum_Price,Sum_Freight_Value,c.state AS State
```

```
FROM DB.mean_ m
```

```
JOIN DB.scode c ON m.customer_state=c.code
```

Query_result:

Row	Mean_price	Mean_freigh...	Sum_Price	Sum_Freigh...	State
1	191.48	42.72	115268.08	25719.73	Paraiba
2	180.89	35.84	80314.81	15914.59	Alagoas
3	173.73	40.07	15982.95	3686.75	Acre
4	165.97	41.07	46140.64	11417.38	Rondonia
5	165.69	35.83	178947.81	38699.3	Para
6	164.32	34.01	13474.3	2788.5	Amapa
7	160.36	39.15	86914.08	21218.2	Piaui
8	157.53	37.25	49621.74	11732.68	Tocantins
9	156.97	35.65	83034.98	18860.1	Rio Grande do Norte
10	153.76	32.71	227254.71	48351.59	Ceara

Insight: Purchase rate of Paraiba is high and least from Ceara out of top 10 records.

5. Analysis on sales, freight and delivery time

i). Calculate days between purchasing, delivering and estimated delivery

ii). Create columns:

time_to_delivery = order_purchase_timestamp - order_delivered_customer_date

diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date

Select c.customer_state AS state,

(sum(distinct ot.freight_value)/count(distinct ot.freight_value)) AS freight_mean ,

(EXTRACT(DAY FROM (EXTRACT(DATE FROM order_delivered_customer_date) - EXTRACT(DATE FROM order_purchase_timestamp)))) AS time_to_delivery ,

(EXTRACT(DAY FROM (EXTRACT(DATE FROM order_estimated_delivery_date) - EXTRACT(DATE FROM order_delivered_customer_date)))) AS diff_estimated_delivery

FROM DB.orders o

join `DB.order_items` ot on ot.order_id=o.order_id

join `DB.customers` c on o.customer_id=c.customer_id

WHERE order_purchase_timestamp is not null and order_delivered_customer_date is not null

iii). Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

Select c.customer_state AS state,

(sum(distinct ot.freight_value)/count(distinct ot.freight_value)) AS freight_mean ,

(EXTRACT(DAY FROM (EXTRACT(DATE FROM order_delivered_customer_date) - EXTRACT(DATE FROM order_purchase_timestamp)))) AS time_to_delivery ,

(EXTRACT(DAY FROM (EXTRACT(DATE FROM order_estimated_delivery_date) - EXTRACT(DATE FROM order_delivered_customer_date)))) AS diff_estimated_delivery

FROM DB.orders o

JOIN `DB.order_items` ot on ot.order_id=o.order_id

JOIN `DB.customers` c on o.customer_id=c.customer_id

WHERE order_purchase_timestamp is not null and order_delivered_customer_date is not null

GROUP BY state, freight_value, time_to_delivery, diff_estimated_delivery

Query_Result:

Row	state	freight_mean	time_to_deli...	diff_estimat...
1	Santa Catarina	18.51	31	29
2	Rio de Janeiro	14.11	36	17
3	Rio Grande do Sul	19.43	31	2
4	Mato Grosso	44.73	33	1
5	Ceara	30.94	44	-4
6	Santa Catarina	19.07	41	-4
7	Pernambuco	35.24	37	-1
8	Rio de Janeiro	15.56	34	-5
9	Alagoas	20.4	39	-6
10	Para	34.15	36	-2

iv). Sort the data to get the following:

i) Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

HIGHEST average freight value

```
Select state FROM (
SELECT c.customer_state AS state,avg(ot.freight_value) AS freight_mean ,
(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_delivered_customer_date) -
EXTRACT(DATE FROM order_purchase_timestamp)))) AS time_to_delivery ,
(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_estimated_delivery_date) -
EXTRACT(DATE FROM order_delivered_customer_date)))) AS diff_estimated_delivery
FROM DB.orders o
join `DB.order_items` ot on ot.order_id=o.order_id
join `DB.customers` c on o.customer_id=c.customer_id
WHERE order_purchase_timestamp is not null and order_delivered_customer_date is not null
GROUP BY state,freight_value,time_to_delivery,diff_estimated_delivery)
ORDER BY freight_mean desc limit 5;
```

Query_Result:

Row	state
1	Piaui
2	Parana
3	Santa Catarina
4	Sao Paulo
5	Mato Grosso

Insight: Piaui is having highest average freight value and Mato Grosso

ii)Top 5 states with highest/lowest average time to delivery

Highest average time to delivery

```
Select distinct state,NTILE(5) over (order by avg_time_to_delivery desc) as dr from(
Select
s.state AS state,
avg(ot.freight_value) AS freight_mean ,
AVG((EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_delivered_customer_date) -
EXTRACT(DATE FROM order_purchase_timestamp)))) AS avg_time_to_delivery,
(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_delivered_customer_date) -
EXTRACT(DATE FROM order_purchase_timestamp)))) AS time_to_delivery ,
```

```

(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_estimated_delivery_date) -
  EXTRACT(DATE FROM order_delivered_customer_date)))) AS diff_estimated_delivery
from DB.orders o
join `DB.order_items` ot on ot.order_id=o.order_id
join `DB.customers` c on o.customer_id=c.customer_id
join `DB.scode` s on s.code=c.customer_state
WHERE order_purchase_timestamp is not null and order_delivered_customer_date is not null
group by state,freight_value,time_to_delivery,diff_estimated_delivery)
order by dr desc limit 5

```

Query_result:

Row	state	dr
1	Pernambuco	5
2	Sergipe	5
3	Alagoas	5
4	Parana	5
5	Maranhao	5

Insight: Pernambuco is having average time to delivery.

Lowest average time to delivery

```

Select distinct state,NTILE(5) over (order by avg_time_to_delivery desc) as dr from(
Select
s.state AS state,
avg(ot.freight_value) AS freight_mean ,
AVG((EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_delivered_customer_date) -
  EXTRACT(DATE FROM order_purchase_timestamp)))) AS avg_time_to_delivery,
(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_delivered_customer_date) -
  EXTRACT(DATE FROM order_purchase_timestamp)))) AS time_to_delivery ,
(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_estimated_delivery_date) -
  EXTRACT(DATE FROM order_delivered_customer_date)))) AS diff_estimated_delivery
from DB.orders o
join `DB.order_items` ot on ot.order_id=o.order_id
join `DB.customers` c on o.customer_id=c.customer_id
join `DB.scode` s on s.code=c.customer_state
WHERE order_purchase_timestamp is not null and order_delivered_customer_date is not null
group by state,freight_value,time_to_delivery,diff_estimated_delivery)
order by dr asc limit 5

```

Query_Result:

Row	state	dr
1	Espirito Santo	1
2	Sergipe	1
3	Roraima	1
4	Alagoas	1
5	Distrito Federal	1

Insight: Espirito Santo is having lowest average time to delivery the product.

iii) Top 5 states where delivery is really fast/ not so fast compared to estimated date

Really fast

```
SELECT distinct state,dr from (
Select distinct state,NTILE(5) over (partition by state order by time_to_delivery) as dr from(
Select
s.state AS state,
avg(ot.freight_value) AS freight_mean ,
(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_delivered_customer_date) -
EXTRACT(DATE FROM order_purchase_timestamp)))) AS time_to_delivery ,
(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_estimated_delivery_date) -
EXTRACT(DATE FROM order_delivered_customer_date)))) AS diff_estimated_delivery
from DB.orders o
join `DB.order_items` ot on ot.order_id=o.order_id
join `DB.customers` c on o.customer_id=c.customer_id
join `DB.scode` s on s.code=c.customer_state
WHERE order_purchase_timestamp is not null and order_delivered_customer_date is not null
group by state,freight_value,time_to_delivery,diff_estimated_delivery)
)order by dr limit 5
```

Query_Result:

Row	state	dr
1	Roraima	1
2	Santa Catarina	1
3	Alagoas	1
4	Pernambuco	1
5	Rondonia	1

Insight: Roraima state is taking very less time to delivery product. It means delivery is very fast.

Slow Delivery:

```
SELECT distinct state,dr from (
Select distinct state,NTILE(5) over (partition by state order by time_to_delivery) as dr from(
Select
s.state AS state,
avg(ot.freight_value) AS freight_mean ,
(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_delivered_customer_date) -
EXTRACT(DATE FROM order_purchase_timestamp)))) AS time_to_delivery ,
(EXTRACT(DAY FROM
(EXTRACT(DATE FROM order_estimated_delivery_date) -
EXTRACT(DATE FROM order_delivered_customer_date)))) AS diff_estimated_delivery
from DB.orders o
join `DB.order_items` ot on ot.order_id=o.order_id
join `DB.customers` c on o.customer_id=c.customer_id
join `DB.scode` s on s.code=c.customer_state
WHERE order_purchase_timestamp is not null and order_delivered_customer_date is not null
group by state,freight_value,time_to_delivery,diff_estimated_delivery)
)order by dr desc limit 5
```

Query_Result:

Row	state	dr
1	Tocantins	5
2	Parana	5
3	Ceara	5
4	Alagoas	5
5	Piaui	5

Insight: Tocantins state's delivery-time of product is very slow

6. Payment type analysis:

i) Month over Month count of orders for different payment types

```
SELECT count(order_id) AS count_order,payment_type,order_delivered_month FROM
(SELECT o.order_id AS order_id,payment_type,
EXTRACT( month FROM CAST(order_delivered_customer_date AS DATE)) AS delivered_month,
format_date('%b',cast(order_delivered_customer_date as DATE)) AS order_delivered_month
FROM `DB.orders` o
join `DB.payments` p ON o.order_id=p.order_id
join `DB.customers` c on c.customer_id=o.customer_id
WHERE order_delivered_customer_date IS NOT NULL) x
group by payment_type,delivered_month,order_delivered_month
order by delivered_month asc,count(order_id) desc
```

Query_Result:

Row	count_order	payment_type	order_delivered_month
1	5211	credit_card	Jan
2	1454	UPI	Jan
3	385	voucher	Jan
4	107	debit_card	Jan
5	5609	credit_card	Feb
6	1425	UPI	Feb
7	412	voucher	Feb
8	79	debit_card	Feb
9	7086	credit_card	Mar
10	1899	UPI	Mar
11	535	voucher	Mar
12	90	debit_card	Mar

Insight: Credit card is the most preferable payment mode in each month and least one is debit card.

Recommendation: Target can collaborate with any bank to offer customer credit with credit points which can be used to shop. In this way both bank and Target will get profit on sales.

ii.) Distribution of payment installments and count of orders

```
SELECT payment_installments, count(distinct order_id) as order_count from
(SELECT o.order_id AS order_id, payment_installments
FROM `DB.orders` o
JOIN `DB.payments` p
ON o.order_id = p.order_id where order_delivered_customer_date is not null)
group by payment_installments
order by count(order_id) desc
```

Query_Result:

Row	payment_installments	payment_mode	order_count
1	1	credit_card	24710
2	1	UPI	19191
3	2	credit_card	12055
4	3	credit_card	10146
5	4	credit_card	6880
6	1	voucher	3679
7	10	credit_card	5137
8	5	credit_card	5090
9	8	credit_card	4122
10	6	credit_card	3800
11	7	credit_card	1560
12	1	debit_card	1484
13	9	credit_card	618
14	12	credit_card	128
15	15	credit_card	72
16	18	credit_card	27
17	11	credit_card	22

Insight: Customers have chosen credit card as payment mode utmost and least debit card for installments.

Recommendation: Target can offer credit card to customer and provide credit points to shop to attract more customers to purchase product.