


TARGET (BRAZIL) – DATA ANALYSIS

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

a. Data type of columns in a table

 **Filter** Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode
<input type="checkbox"/>	order_id	STRING	NULLABLE
<input type="checkbox"/>	customer_id	STRING	NULLABLE
<input type="checkbox"/>	order_status	STRING	NULLABLE
<input type="checkbox"/>	order_purchase_timestamp	TIMESTAMP	NULLABLE
<input type="checkbox"/>	order_approved_at	TIMESTAMP	NULLABLE
<input type="checkbox"/>	order_delivered_carrier_date	TIMESTAMP	NULLABLE
<input type="checkbox"/>	order_delivered_customer_date	TIMESTAMP	NULLABLE
<input type="checkbox"/>	order_estimated_delivery_date	TIMESTAMP	NULLABLE

b. Time period for which the data is given.

This business case has information of 100k orders **from 2016 to 2018** made at Target in Brazil .

c. Cities and States of customers ordered during the given period.

```
select distinct c.customer_city, c.customer_state, o.order_purchase_timestamp, c.customer_id
from `target_sql.orders` o
join `target_sql.customers` c
on o.customer_id = c.customer_id
```

Row	customer_city	customer_state	order_purchase_timestamp	customer_id
1	maceio	AL	2017-07-11 13:36:30 UTC	5fc4c97dcb63903f996714524...
2	aracaju	SE	2018-07-11 20:24:49 UTC	a5c8228ef32a5a250903b18c0...
3	aracaju	SE	2018-06-23 13:25:15 UTC	670af30ca5b8c20878fecdfa5...
4	maceio	AL	2017-07-30 16:45:22 UTC	5351c1e4ae199735063d6406c...
5	teresina	PI	2017-02-24 13:50:32 UTC	5b54155ba8103b1bb1e157ed...
6	pau d'arco	AL	2018-03-21 23:45:08 UTC	1318775058e4321f5018e2fe4...
7	natal	RN	2018-06-13 09:52:02 UTC	9c4efecd1866c2177998d461b...
8	teresina	PI	2018-05-18 23:21:25 UTC	84cb4824ee3f6d0c24b60d12a...
9	sao joao do piaui	PI	2018-04-25 10:38:57 UTC	6143e5df1b61e9568a5f02adb...
10	boquim	SE	2018-06-18 21:09:41 UTC	de270dbea5d94e6436d84456...

2. In-depth Exploration:

- 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?

Yes, the e-commerce trend is growing . From the starting year of 2016 the no of purchases increases drastically. And at some specific months the purchases are very high due to seasonality.

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

Row	year	month	no_of_purchases
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

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

Mostly during night.

```

select extract(hour from order_purchase_timestamp) as hour,
count(order_id ) as no_of_purchases
from `target_sql.orders`
group by hour
order by hour

```

Row	hour	no_of_purchases
1	0	2394
2	1	1170
3	2	510
4	3	272
5	4	206
6	5	188
7	6	502
8	7	1231

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

a. Get month on month orders by states

```

select count (distinct o.order_id) as total_orders,g.geolocation_state as state,
extract(month from o.order_purchase_timestamp) month,
extract(year from o.order_purchase_timestamp) year
from `target_sql.orders` o
inner join `target_sql.customers` c
on o.customer_id = c.customer_id
inner join `target_sql.geolocation` g
on c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
group by g.geolocation_state,month,year
order by year,month,state

```

Row	total_orders	state	month	year
1	1	RR	9	2016
2	1	RS	9	2016
3	2	SP	9	2016
4	2	AL	10	2016
5	4	BA	10	2016
6	8	CE	10	2016
7	6	DF	10	2016
8	4	ES	10	2016
9	9	GO	10	2016

b. Distribution of customers across the states in Brazil

```
select count(distinct c.customer_unique_id) cust_count,
b.geolocation_state state from `target_sql.customers` c
join `target_sql.geolocation` g on
c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
group by g.geolocation_state
order by cust_count desc
```

Row	cust_count	state
1	40287	SP
2	12372	RJ
3	11248	MG
4	5284	RS
5	4871	PR
6	3547	SC
7	3268	BA
8	1959	ES
9	1944	GO

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

1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value" column in payments table

```
with base1 as (
  select * from (
    select EXTRACT(YEAR from order_purchase_timestamp) as year,
    EXTRACT(MONTH from order_purchase_timestamp) as month,
    sum(payment_value) as total
    from `target_sql.orders` o inner join `target_sql.payments` p
    on o.order_id=p.order_id
    group by year, month
    order by year, month) temp
  where temp.month in (1,2,3,4,5,6,7,8) and temp.year in(2017,2018)),
base2 as (
  select year,round(sum(total)) as sum_of_total from base1 group by year)
  select *,(sum_of_total- lag(sum_of_total)
over(order by year))/sum_of_total*100 as per_increase
from base2
```

Row	year	sum_of_total	per_increase
1	2018	8694734.0	57.8017912...
2	2017	3669022.0	null

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```
select
-1*date_diff(o.order_purchase_timestamp,o.order_delivered_customer_date,day)
as time_to_delivery,
date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)
as diff_estimated_delivery,o.order_purchase_timestamp,o.order_delivered_customer_date,
o.order_estimated_delivery_date
from `target_sql.orders`o
inner join `target_sql.order_items`oi
on o.order_id = oi.order_id
join `target_sql.customers` c
on c.customer_id = o.customer_id
where o.order_purchase_timestamp is not null and
o.order_delivered_customer_date is not null and
o.order_estimated_delivery_date is not null
```

Row	time_to_delivery	diff_estimated_delivery	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date
1	30	-12	2018-02-19 19:48:52 UTC	2018-03-21 22:03:51 UTC	2018-03-09 00:00:00 UTC
2	30	28	2016-10-09 15:39:56 UTC	2016-11-09 14:53:50 UTC	2016-12-08 00:00:00 UTC
3	35	16	2016-10-03 21:01:41 UTC	2016-11-08 10:58:34 UTC	2016-11-25 00:00:00 UTC
4	30	1	2017-04-15 15:37:38 UTC	2017-05-16 14:49:55 UTC	2017-05-18 00:00:00 UTC
5	32	0	2017-04-14 22:21:54 UTC	2017-05-17 10:52:15 UTC	2017-05-18 00:00:00 UTC
6	32	0	2017-04-14 22:21:54 UTC	2017-05-17 10:52:15 UTC	2017-05-18 00:00:00 UTC
7	29	1	2017-04-16 14:56:13 UTC	2017-05-16 09:07:47 UTC	2017-05-18 00:00:00 UTC
8	43	-4	2017-04-08 21:20:24 UTC	2017-05-22 14:11:31 UTC	2017-05-18 00:00:00 UTC
9	40	-4	2017-04-11 19:49:45 UTC	2017-05-22 16:18:42 UTC	2017-05-18 00:00:00 UTC

2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- $\text{time_to_delivery} = \text{order_purchase_timestamp} - \text{order_delivered_customer_date}$
- $\text{diff_estimated_delivery} = \text{order_estimated_delivery_date} - \text{order_delivered_customer_date}$

```
select
-1*date_diff(o.order_purchase_timestamp,o.order_delivered_customer_date,day)
as time_to_delivery,
date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)
as diff_estimated_delivery
from `target_sql.orders`o
inner join `target_sql.order_items`oi
on o.order_id = oi.order_id
join `target_sql.customers` c
on c.customer_id = o.customer_id
where o.order_purchase_timestamp is not null and
o.order_delivered_customer_date is not null and
o.order_estimated_delivery_date is not null
```

Row	time_to_delivery	diff_estimated_c
1	30	-12
2	30	28
3	35	16
4	30	1
5	32	0
6	32	0
7	29	1
8	43	-4
9	40	-4

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

```
with demo as (
select oi.freight_value as fv, c.customer_state as state,
-1*date_diff(o.order_purchase_timestamp,o.order_delivered_customer_date,day) as time_to_delivery,
-1*date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)
as diff_estimated_delivery,o.order_purchase_timestamp,o.order_delivered_customer_date,
o.order_estimated_delivery_date
from `target_sql.orders`o
inner join `target_sql.order_items`oi
on o.order_id = oi.order_id
join `target_sql.customers` c
on c.customer_id = o.customer_id
where o.order_purchase_timestamp is not null
and o.order_delivered_customer_date is not null and
o.order_estimated_delivery_date is not null)
select avg(d.fv) as avg_FV,avg(d.time_to_delivery) as avg_Time_To_Delivery,
avg(d.diff_estimated_delivery) as avg_diff_estimated_delivery,
d.state
from demo d
group by d.state
```

Row	avg_FV	avg_Time_To_De	avg_diff_estimated_d	state
1	20.9097843...	14.6893821...	-11.14449314293...	RJ
2	20.6258372...	11.5155221...	-12.39715104126...	MG
3	21.5066276...	14.5209858...	-10.66886285993...	SC
4	15.1149940...	8.25960855...	-10.26559438451...	SP
5	22.5628678...	14.9481774...	-11.37285902503...	GO
6	21.6142703...	14.7082993...	-13.20300016305...	RS
7	26.4875563...	18.7746402...	-10.11946782514...	BA
8	27.9969141...	17.5081967...	-13.63934426229...	MT
9	36.5731733...	20.9786666...	-9.1653333333333...	SE
10	32.6933333...	17.7920962...	-12.55211912943...	PE

Sort the data to get the following:

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

```
with demo as ([
select oi.freight_value as fv, c.customer_state as state, -1*date_diff(o.order_purchase_timestamp,o.
order_delivered_customer_date,day) as time_to_delivery,
-1*date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery,o.
order_purchase_timestamp,o.order_delivered_customer_date,o.order_estimated_delivery_date
from `target_sql.orders` o
inner join `target_sql.order_items` oi
on o.order_id = oi.order_id
join `target_sql.customers` c
on c.customer_id = o.customer_id
where o.order_purchase_timestamp is not null and o.order_delivered_customer_date is not null and o.
order_estimated_delivery_date is not null])

select avg(d.fv) as avg_FV--,avg(d.time_to_delivery) as avg_Time_To_Delivery, avg(d.diff_estimated_delivery) as
avg_diff_estimated_delivery
,d.state
from demo d
group by d.state
order by avg_FV asc
limit 5

# change asc to desc for bottom 5 results
```

Row	avg_FV	state
1	15.1149940...	SP
2	20.4718162...	PR
3	20.6258372...	MG
4	20.9097843...	RJ
5	21.0721613...	DF

Row	avg_FV	state
1	43.0916894...	PB
2	43.0880434...	RR
3	41.3305494...	RO
4	40.0479120...	AC
5	39.1150860...	PI

6. Top 5 states with highest/lowest average time to delivery

```

with demo as (
select oi.freight_value as fv, c.customer_state as state, -1*date_diff(o.order_purchase_timestamp,o.
order_delivered_customer_date,day) as time_to_delivery,
-1*date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery,o.
order_purchase_timestamp,o.order_delivered_customer_date,o.order_estimated_delivery_date
from `target_sql.orders`o
inner join `target_sql.order_items`oi
on o.order_id = oi.order_id
join `target_sql.customers` c
on c.customer_id = o.customer_id
where o.order_purchase_timestamp is not null and o.order_delivered_customer_date is not null and o.
order_estimated_delivery_date is not null)

select
avg(d.time_to_delivery) as avg_Time_To_Delivery
,d.state
from demo d
group by d.state
order by avg_Time_To_Delivery desc
limit 5

# change asc to desc for bottom 5 results

```

Row	avg_Time_To_Delivery	state
1	8.25960855...	SP
2	11.4807930...	PR
3	11.5155221...	MG
4	12.5014861...	DF
5	14.5209858...	SC

Row	avg_Time_To_Delivery	state
1	27.8260869...	RR
2	27.7530864...	AP
3	25.9631901...	AM
4	23.9929742...	AL
5	23.3017077...	PA

7. Top 5 states where delivery is really fast/ not so fast compared to estimated date


```

with demo as (
select oi.freight_value as fv, c.customer_state as state, -1*date_diff(o.order_purchase_timestamp,o.
order_delivered_customer_date,day) as time_to_delivery,
date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery,o.
order_purchase_timestamp,o.order_delivered_customer_date,o.order_estimated_delivery_date
from `target_sql.orders` o
inner join `target_sql.order_items` oi
on o.order_id = oi.order_id
join `target_sql.customers` c
on c.customer_id = o.customer_id
where o.order_purchase_timestamp is not null and o.order_delivered_customer_date is not null and o.
order_estimated_delivery_date is not null)

select
avg(d.diff_estimated_delivery) as avg_diff_estimated_delivery,d.state
from demo d
group by d.state
order by avg_diff_estimated_delivery
limit 5

# change asc to desc for bottom 5 results

```

Row	avg_diff_estimated_delivery	state
1	7.97658079...	AL
2	9.10999999...	MA
3	9.16533333...	SE
4	9.76853932...	ES
5	10.1194678...	BA

Row	avg_diff_estimated_delivery	state
1	20.0109890...	AC
2	19.0805860...	RO
3	18.9754601...	AM
4	17.4444444...	AP
5	17.4347826...	RR

6. Payment type analysis:

- Month over Month count of orders for different payment types

```

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

```

Row	year	month	no_of_purchases	payment_type
1	2017	1	197	UPI
2	2017	1	583	credit_card
3	2017	1	9	debit_card
4	2017	1	61	voucher
5	2018	1	1518	UPI
6	2018	1	5520	credit_card
7	2018	1	109	debit_card
8	2018	1	416	voucher
9	2017	2	398	UPI

2.Count of orders based on the no. of payment installments

```
select payment_installments, count(order_id) as orders,
from `target_sql.payments`
group by payment_installments
order by orders desc
```

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

Insights –

1.

- The e-commerce trend is growing every year and at some specific months the purchases are high due to seasonality.
- More no of purchases are made during the time from 10 am to 11pm

2. No of orders and customers are high in specific states

3.

- Time to delivery takes longer than the estimated time period (average of 30 days)
- The average freight time takes 20 to 30 days depending on the states
- The delivery periods are faster in some states and slower in others

4.

- Most no of purchases were made with credit card and UPI and less with vouchers.
- High no of people prefers paying in single installments.

RECOMMENDATIONS -

1. Since the e-commerce trend is growing stock more no of products every year and also during specific peak hours .

2. In the states where it has the high no of orders and customers , hire extra employees for handling the customers needs and orders smoothly.

3. Hire more no of delivery drivers to get faster to get faster order delivery record and freight time (This in turn helps with higher rate of customer satisfaction and gets goods to the store much faster)

4. Gift vouchers can be given to increase the no of purchases using vouchers . And ZERO – COST EMI can be implemented to increase more no of customers to purchase products .