

Business Case Study – Target

Done by
Preethi Tony

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.

<input type="checkbox"/>	Field name	Type	Mode
<input type="checkbox"/>	customer_id	STRING	NULLABLE
<input type="checkbox"/>	customer_unique_id	STRING	NULLABLE
<input type="checkbox"/>	customer_zip_code_prefix	INTEGER	NULLABLE
<input type="checkbox"/>	customer_city	STRING	NULLABLE
<input type="checkbox"/>	customer_state	STRING	NULLABLE

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

```
SELECT order_status,MIN(order_purchase_timestamp) AS start_time,  
MAX(order_purchase_timestamp) AS end_time,  
COUNT(*) AS total_orders  
FROM `Target.orders`  
WHERE  
    order_purchase_timestamp >= '2016-01-01'  
    AND order_purchase_timestamp <= '2018-12-31'  
    and order_status = "created"  
group by order_status
```

Row	order_status	start_time	end_time	total_orders
1	created	2017-11-06 13:12:34 UTC	2018-02-09 17:21:04 UTC	5

Insights: We have customers who have placed order between the timestamp 2017-11-06 to 2018-02-09 and total of 5 orders were created in the time period

3.Count the Cities & States of customers who ordered during the given period

```
select count(c.customer_city) as city_count,count(c.customer_state) as state_count  
from `Target.customers` c inner join `Target.orders` o  
on c.customer_id=o.customer_id  
where o.order_id is not null
```

Row	city_count	state_count
1	99441	99441

```
select customer_city,max(city_count) as maximum_order_city,customer_state,max(state_count) as
maximum_order_state from (select c.customer_city,c.customer_state, count(c.customer_city) as
city_count,count(c.customer_state) as state_count
from `Target.customers` c inner join `Target.orders` o
on c.customer_id=o.customer_id
where o.order_id is not null
group by c.customer_city,c.customer_state)t
group by customer_city,customer_state
order by maximum_order_city desc,maximum_order_state desc
```

Row	customer_city	maximum_order_city	customer_state	maximum_order_state
1	sao paulo	15540	SP	15540
2	rio de janeiro	6882	RJ	6882
3	belo horizonte	2773	MG	2773
4	brasil	2131	DF	2131
5	curitiba	1521	PR	1521
6	campinas	1444	SP	1444
7	porto alegre	1379	RS	1379
8	salvador	1245	BA	1245
9	guarulhos	1189	SP	1189
10	sao bernardo do campo	938	SP	938

Insights: Between 2016 and 2018 we have exactly **99,441** customers who have placed order. The most placed order is from **"Sao Paulo"** with maximum count of **15,540**.

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 order_year,
count(*) as order_count
from `Target.orders`
group by order_year
order by order_year desc
```

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

Insights: We can see there is a growing trend from 2016-2018, In **2016** we had only **329** orders placed and later in **2017** we had **45101** orders placed and in the year **2018** we have **54011** orders placed. We have seen the graph in the positive direction.

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 order_month,
extract(year from order_purchase_timestamp) as order_year,
count(order_id) as order_count
from `Target.orders`
group by order_year,order_month
order by order_year desc,order_month desc;
```

Row	order_month	order_year	order_count
1	10	2018	4
2	9	2018	16
3	8	2018	6512
4	7	2018	6292
5	6	2018	6167
6	5	2018	6873
7	4	2018	6939
8	3	2018	7211
9	2	2018	6728
10	1	2018	7269
11	12	2017	5673
12	11	2017	7544
13	10	2017	4631
14	9	2017	4285
15	8	2017	4331
16	7	2017	4026
17	6	2017	3245
18	5	2017	3700
19	4	2017	2404
20	3	2017	2682
21	2	2017	1780
22	1	2017	800
23	12	2016	1
24	10	2016	324
25	9	2016	4

Insights: Looks like there is a variation in the orders placed. If we investigate the **October** month orders for all three years, we are getting the following information.

2016 – 324

2017 – 4631

2018 – 4.

They graph shows variation and there is no particular reason for the same.

3.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

```

select count(order_id) as Total_count,case
when order_hours between 0 and 6 then "Dawn"
when order_hours between 7 and 12 then "Morning"
when order_hours between 13 and 18 then "Afternoon"
else "Night"
end as Schedule from
(select order_id,extract(hour from order_purchase_timestamp) as order_hours
from `Target.orders`)t
group by Schedule

```

Row	Total_count	Schedule
1	27733	Morning
2	5242	Dawn
3	38135	Afternoon
4	28331	Night

Insights: Most of the orders were placed in the Afternoon and the second most is in Night

Evolution of E-commerce orders in the Brazil region:

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

Assumptions: By calculating the number of orders placed in each state for each month, we can identify if there are any patterns or fluctuations in the order volumes on a monthly basis. This analysis can provide valuable insights into the purchasing behaviour of customers in different states and help the company to make data-driven decisions related to inventory management, marketing strategies, and resource allocation.

```

SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,customer_state,
COUNT(order_id) AS order_count
FROM `Target.orders` AS o JOIN `Target.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY order_year, order_month, customer_state
ORDER BY order_count desc

```

Row	order_year	order_month	customer_state	order_count
1	2018	8	SP	3253
2	2018	5	SP	3207
3	2018	4	SP	3059
4	2018	1	SP	3052
5	2018	3	SP	3037
6	2017	11	SP	3012
7	2018	7	SP	2777
8	2018	6	SP	2773
9	2018	2	SP	2703
10	2017	12	SP	2357
11	2017	10	SP	1793
12	2017	8	SP	1729
13	2017	9	SP	1638
14	2017	7	SP	1604
15	2017	5	SP	1425
16	2017	6	SP	1331
17	2017	11	RJ	1048
18	2017	3	SP	1010
19	2017	11	MG	943
20	2018	2	RJ	922
21	2017	4	SP	908
22	2018	3	RJ	907
23	2018	1	RJ	893
24	2018	3	MG	879
25	2018	1	MG	863

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
11	2016	10	MG	40
12	2016	10	MT	3
13	2016	10	PA	4
14	2016	10	PB	1
15	2016	10	PE	7
16	2016	10	PI	1
17	2016	10	PR	19
18	2016	10	RJ	56
19	2016	10	RN	4
20	2016	10	RR	1
21	2016	10	RS	24
22	2016	10	SC	11
23	2016	10	SE	3
24	2016	10	SP	113
25	2016	12	PR	1

Insights: Customers have started placing order from **Sept,2019**. The most placed order is from the state **SP** and the total **No.Of.Orders** is **3253** in the month of **Aug,2018**.

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

```
select customer_state,count(customer_state) as state_count
from `Target.customers`
group by customer_state
order by state_count desc
```

Row	customer_state	state_count
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
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

Insights Customers are distributed across 27 states and customers from **SP** state is the **highest**.

Impact on Economy

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

WITH year_totals AS

```
(
  SELECT EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
         extract(month from o.order_purchase_timestamp) as order_month,
         round(SUM(p.payment_value),2) AS total_value
  FROM `Target.orders` o join `Target.payments` p
  on o.order_id=p.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 order_year,order_month
)
SELECT y2017.order_year as order_year_2017,y2018.order_year as
order_year_2018,y2017.order_month as order_month_2017,y2018.order_month as
order_month_2018,y2017.total_value as total_value_2017,y2018.total_value as total_value_2018,
round((y2018.total_value - y2017.total_value) / ABS(y2017.total_value) * 100,2) AS
percentage_increase
FROM year_totals y2017 JOIN year_totals y2018
ON y2017.order_year = 2017 AND y2018.order_year = 2018
where y2017.order_year between 2017 and 2018 and
y2018.order_month between 1 and 8
order by y2017.order_year,y2018.order_year,y2017.order_month,y2018.order_month;
```

Row	order_year_2017	order_year_2018	order_month_2017	order_month_2018	total_value_2017	total_value_2018	percentage_increase
1	2017	2018	1	1	138488.04	1115004.18	705.13
2	2017	2018	1	2	138488.04	992463.34	616.64
3	2017	2018	1	3	138488.04	1159652.12	737.37
4	2017	2018	1	4	138488.04	1160785.48	738.18
5	2017	2018	1	5	138488.04	1153982.15	733.27
6	2017	2018	1	6	138488.04	1023880.5	639.33
7	2017	2018	1	7	138488.04	1066540.75	670.13
8	2017	2018	1	8	138488.04	1022425.32	638.28
9	2017	2018	2	1	291908.01	1115004.18	281.97
10	2017	2018	2	2	291908.01	992463.34	239.99

Insights: Percentage increase in cost of orders from 2017 and 2018 reflects the economic growth occurred. At times seasonal trends and customer segments impact variation in the trend.

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

```
select distinct customer_state, total_amount, avg_amount from (select c.customer_state,
round(sum(p.payment_value) over(partition by c.customer_state order by c.customer_state),2) as
total_amount,
round(avg(p.payment_value) over(partition by c.customer_state order by c.customer_state),2) as
avg_amount
from `Target.customers` c join `Target.orders` o
on c.customer_id=o.customer_id
join `Target.payments` p
on o.order_id=p.order_id
group by c.customer_state, p.payment_value)t
order by customer_state
```

Row	customer_state	total_amount	avg_amount
1	AC	19533.03	235.34
2	AL	91913.07	229.78
3	AM	27697.59	183.43
4	AP	16191.66	234.66
5	BA	523152.83	197.04
6	CE	255021.6	217.04
7	DF	308090.24	178.4
8	ES	287581.48	170.77
9	GO	312960.11	182.59
10	MA	142391.3	209.09
11	MG	1320753.13	189.52
12	MS	129368.65	194.83
13	MT	177301.57	204.26
14	PA	207614.76	230.68
15	PB	133624.05	257.96

Row	customer_state	total_amount	avg_amount
1	SP	3304585.03	203.23
2	RJ	1499773.44	199.15
3	MG	1320753.13	189.52

Insights: As per the data State **SP** has the Highest amount of **33,04,585.03** and the Average amount is **203.23**

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

```
select distinct customer_state, Total_freight_value, Avg_freight_value from
(select c.customer_state, round(sum(oi.freight_value) over(partition by c.customer_state order by
c.customer_state),2) as Total_freight_value,
round(avg(oi.freight_value) over(partition by c.customer_state order by c.customer_state),2) as
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, oi.freight_value)t
order by customer_state
```

Row	customer_state	Total_freight_value	Avg_freight_value
1	AC	3078.18	42.75
2	AL	12031.87	38.69
3	AM	4065.5	35.35
4	AP	2282.73	36.23
5	BA	47819.59	34.11
6	CE	30658.4	38.32
7	DF	25772.03	26.6
8	ES	25575.68	27.47
9	GO	27620.61	28.74
10	MA	20819.56	41.81
11	MG	77356.18	30.81
12	MS	12485.4	28.38
13	MT	19356.09	32.7
14	PA	26922.12	40.79
15	PB	18009.12	45.25
Row	customer_state	Total_freight_value	Avg_freight_value
1	SP	115342.71	29.91
2	RJ	86061.41	31.61
3	MG	77356.18	30.81

Insights: As per the data State **SP** has the Highest freight value of **1,15,342.71** and the Average freight value is **29.91**

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.

```
select * from(select order_id,  
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_took_deliver,  
timestamp_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as  
estimated_delivery_time  
from `Target.orders`)t  
where time_to_deliver is not null
```

Row	order_id	time_took_deliver	estimated_delivery_t
1	1950d777989f6a877539f5379...	30	17
2	2c45c33d2f9cb8ff8b1c86cc28...	30	59
3	65d1e226dfaeb8cdc42f66542...	35	52
4	635c894d068ac37e6e03dc54e...	30	32
5	3b97562c3aee8bdedcb5c2e45...	32	33
6	68f47f50f04c4cb6774570cfde...	29	31
7	276e9ec344d3bf029ff83a161c...	43	39
8	54e1a3c2b97fb0809da548a59...	40	36
9	fd04fa4105ee8045f6a0139ca5...	37	35
10	302bb8109d097a9fc6e9cefc5...	33	28

Insights: From the report it is upsetting to know that the Time took to deliver is greater than the estimated delivery time.

Thoughts: If we hire more logistic partner to deliver our orders, we can speedup the process in delivering the time within the estimated period itself.

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

```
select c.customer_state,round(avg(oi.freight_value),0) as 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;
```

Row	customer_state	Avg_freight_value
1	SP	15.0
2	PR	21.0
3	RJ	21.0
4	DF	21.0
5	MG	21.0

```
select c.customer_state,round(avg(oi.freight_value),0) as 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;
```

Row	customer_state	Avg_freight_value
1	PB	43.0
2	RR	43.0
3	RO	41.0
4	AC	40.0
5	PI	39.0

Insights: States **SP, PR, RJ, DF, MG** have charged lowest freight value in average and States **PB, RR, RO, AC, PI** have charged highest Freight value in average

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

```
select customer_state,round(avg(difference),0) as Avg_difference from
(select c.customer_state,
timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as difference
from `Target.orders` o join `Target.customers` c
on o.customer_id=c.customer_id)t
group by customer_state
order by Avg_difference
limit 5;
```

Row	customer_state	Avg_difference
1	SP	8.0
2	MG	12.0
3	PR	12.0
4	DF	13.0
5	SC	14.0

```

select customer_state,round(avg(difference),0) as Avg_difference from
(select c.customer_state,
timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as difference
from `Target.orders` o join `Target.customers` c
on o.customer_id=c.customer_id)t
group by customer_state
order by Avg_difference desc
limit 5;

```

Row	customer_state	Avg_difference
1	RR	29.0
2	AP	27.0
3	AM	26.0
4	AL	24.0
5	PA	23.0

Insights: We have states **SP, MG, PR, DF, SC** had lowest and was faster in delivering the products wherein we have states **RR, AP, AM, AL, PA** had highest and took long time to deliver the products

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

```

select distinct customer_state,t.time_took_deliver,t.estimated_delivery_time from(select
c.customer_state,o.order_status,
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_took_deliver,
timestamp_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as
estimated_delivery_time
from `Target.orders` o join `Target.customers` c
on o.customer_id=c.customer_id)t
where time_took_deliver is not null and time_took_deliver < estimated_delivery_time and order_status =
"delivered"
order by t.time_took_deliver desc
limit 5

```

Row	customer_state	time_took_deliver	estimated_delivery_t
1	CE	70	94
2	SC	63	101
3	MG	60	63
4	SC	59	71
5	SP	55	96

Insights: States **CE, SC, MG, SC, SP** have done the fastest delivery before the estimated delivery period

Analysis based on the payments:

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

```
select extract(month from o.order_purchase_timestamp) as order_month,p.payment_type,count(*) as  
no_of_orders  
from `Target.orders` o join `Target.payments` p  
on o.order_id=p.order_id  
group by order_month,p.payment_type  
order by order_month
```

Row	order_month	payment_type	no_of_orders
1	1	credit_card	6103
2	1	UPI	1715
3	1	voucher	477
4	1	debit_card	118
5	2	UPI	1723
6	2	credit_card	6609
7	2	voucher	424
8	2	debit_card	82
9	3	credit_card	7707
10	3	UPI	1942
11	3	debit_card	109
12	3	voucher	591
13	4	voucher	572
14	4	credit_card	7301
15	4	UPI	1783

Insights: From the following data below is the complete finding
Total no.of.orders made using **credit card** is **76795**
Total no.of.orders made using **UPI** is **19784**
Total no.of.orders made using **voucher** is **5775**
Total no.of.orders made using **debit card** is **1465**
Total no.of.orders made that is **not defined** is **3**

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

```
select extract(month from o.order_purchase_timestamp) as order_month,p.payment_type,count(*) as  
no_of_orders  
from `Target.orders` o join `Target.payments` p  
on o.order_id=p.order_id  
group by order_month,p.payment_type
```

order by order_month

Row	payment_installment	total_count ▼
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644
11	10	5328
12	11	23
13	12	133
14	13	16
15	14	15

Insights: Customers have made instant payments and the **payment instalments** was up to 24 months. comparatively most of the customers have made payments within 10 months itself, only few were making payments up to 24 months.