Target case study

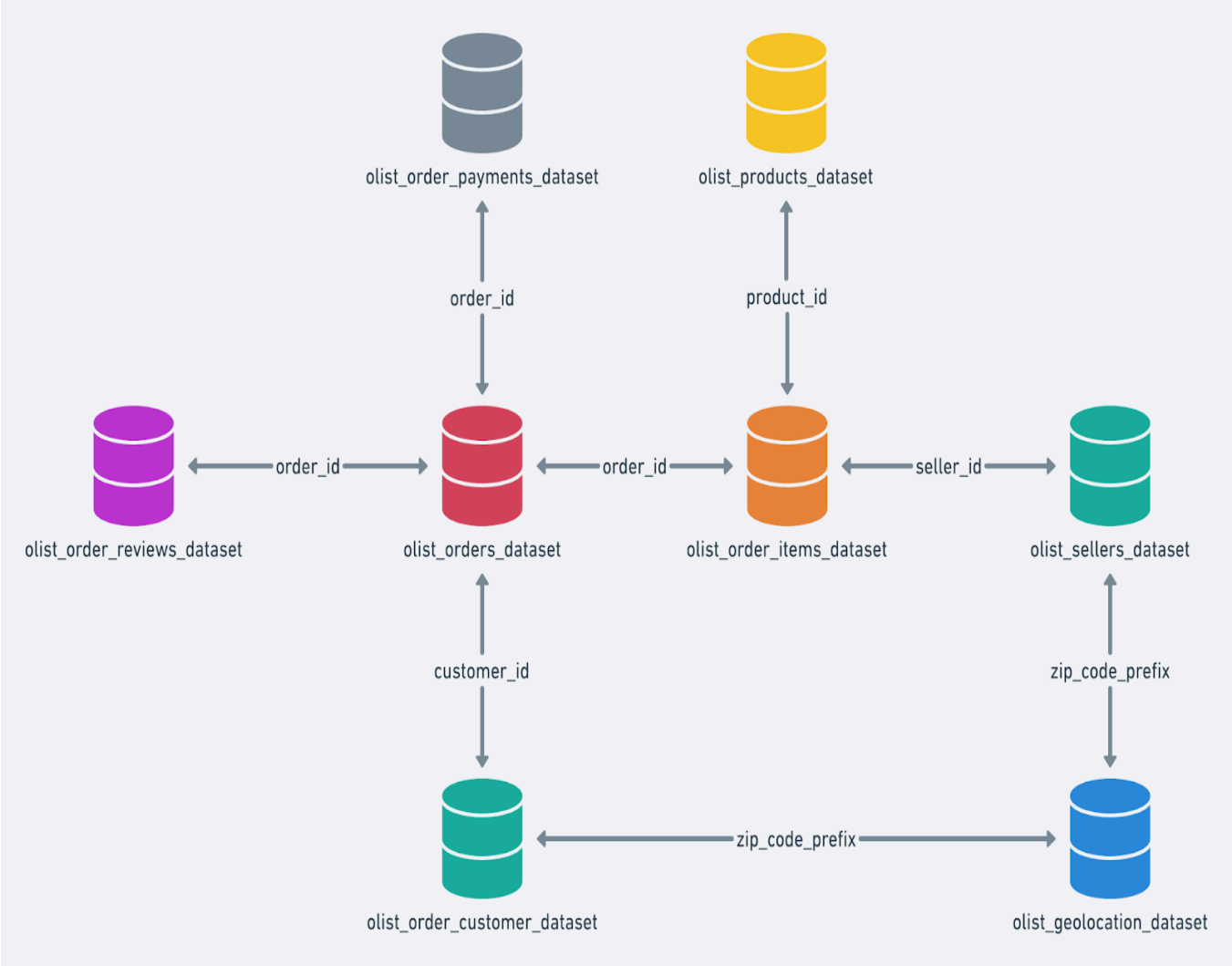
About Target

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

Objective

This particular dataset is from a brazil operation and we have to analyze the given dataset to extract valuable insights and provide actionable recommendations.

Dataset schema



1. Checking the structure and characteristic of dataset

- **Data type of all columns in the "customers" table.**

```
SELECT column_name, data_type
FROM `target_sql_dsml.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

NOTE : - information_schema is a special function in SQL that stores metadata about the database

- **Time range in which the orders were placed**

```
select MIN(date(order_purchase_timestamp)) as first_order ,
MAX(date(order_purchase_timestamp)) as last_order
from `target_sql_dsml.orders`
```

Row	first_order	last_order
1	2016-09-04	2018-10-17

- **Counting the cities and states of customers who ordered during the given period**

```
SELECT COUNT(DISTINCT customers.customer_city) as no_of_city,
COUNT(DISTINCT customers.customer_state) as no_of_state
FROM `target_sql_dsml.orders` AS orders
LEFT JOIN `target_sql_dsml.customers` AS customers
ON orders.customer_id = customers.customer_id
```

```
WHERE orders.order_purchase_timestamp BETWEEN '2016-09-04' AND
'2018-10-17';
```

Row	no_of_city	no_of_state
1	4119	27

2. In depth exploration

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

```
with cte1 as
( select EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year
, count(*) as current_year_order_count
from `target_sql_dsml.orders`
group by EXTRACT(YEAR FROM order_purchase_timestamp)
order by order_year),
cte2 as (
 select * , lag(current_year_order_count) over (order by
order_year ASC) as previous_year_order_value
 from cte1
)
select
*,round((current_year_order_count/previous_year_order_value),2) as
rate_of_increase_or_decrease
from cte2
order by order_year
```

Row	order_year	current_year_order_c	previous_year_order_	rate_of_increase_or_
1	2016	329	*null*	*null*
2	2017	45101	329	137.09
3	2018	54011	45101	1.2

Insight : as we can see from the table that there is an increase in no. orders from 2016 to 2017 of about 137.09 and there is also an increase in order from 2017 to 2018 of about 1.2 which

means that there is huge growth from 2016 to 2017 which is likely due to introduction or early adoption phase of the service where a sudden surge came in 2017 , the subsequent increase in 2017 to 2018 indicates a more stable growth .

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

```
WITH cte1 AS (
 SELECT
   EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
   EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
   COUNT(*) AS order_count
 FROM `target_sql_dsml.orders`
 GROUP BY year, month
 ORDER BY year, month
),
cte2 AS (
 SELECT *,
        FORMAT_DATE('%B', DATE(year, month, 1)) AS month_name
 FROM cte1
),cte3 as (
SELECT year , month_name , order_count , lag(order_count)over
(partition by year order by month) as previous_month_order_count
FROM cte2)
select * , round(order_count/previous_month_order_count , 2) as
rate_of_increase_or_decrease
from cte3
order by year
```

Row	month_name	order_count	previous_month_ord	rate_of_increase_or
1 2016	September	4	*null*	*null*
2 2016	October	324	4	81.0
3 2016	December	1	324	0.0
4 2017	January	800	*null*	*null*
5 2017	February	1780	800	2.23
6 2017	March	2682	1780	1.51
7 2017	April	2404	2682	0.9
8 2017	May	3700	2404	1.54
9 2017	June	3245	3700	0.88
10 2017	July	4026	3245	1.24
11 2017	August	4331	4026	1.08
12 2017	September	4285	4331	0.99
13 2017	October	4631	4285	1.08
14 2017	November	7544	4631	1.63
15 2017	December	5673	7544	0.75

Insights : from this analysis we can see that there is an increase in the number of people going to store during feb,march , jun , oct, nov this could be due to the summer season shopping , june marks the beginning of winter season in brazil , festivals , early christmas preparation.

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

```
with cte1 as
(
 select order_id , extract(hour from order_purchase_timestamp) as
hour_of_purchase
 from `target_sql_dsml.orders`
), cte2 as
(
 select *, case
when hour_of_purchase between 0 and 6 then 'Dawn'
when hour_of_purchase between 7 and 12 then 'Mornings'
when hour_of_purchase between 13 and 18 then 'Afternoon'
when hour_of_purchase between 19 and 23 then 'Night'
end as time_of_day
from cte1
```

```
)
select time_of_day , count(*) as total_orders
from cte2
group by time_of_day
order by total_orders DESC
limit 1
```

Row	time_of_day	total_orders
1	Afternoon	38135

INSIGHTS : we can clearly see that people love to go shopping in the afternoons . Brazil has a tropical climate with high temperatures in the midday and often it reaches uncomfortable levels, however in the afternoon after the peak heat the weather can become more pleasant, encouraging people to head to stores. And it can also be the case that people love to go out with their families when their children are back from school.

3. Evolution of E-commerce orders in the brazil region

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

```
with cte1 as
(
 select o.order_id , c.customer_state , extract(month from
order_purchase_timestamp) as month
 from `target_sql_dsml.customers` as c
 join `target_sql_dsml.orders` as o
 on c.customer_id = o.customer_id
```

```sql
),cte2 as
(
 select customer_state , month , count(*) as total_orders
 from cte1
 group by customer_state , month
 order by customer_state , month
)
select customer_state ,
CASE
   WHEN month = 1 THEN 'Jan'
   WHEN month = 2 THEN 'Feb'
   WHEN month = 3 THEN 'Mar'
   WHEN month = 4 THEN 'Apr'
   WHEN month = 5 THEN 'May'
   WHEN month = 6 THEN 'Jun'
   WHEN month = 7 THEN 'Jul'
   WHEN month = 8 THEN 'Aug'
   WHEN month = 9 THEN 'Sep'
   WHEN month = 10 THEN 'Oct'
   WHEN month = 11 THEN 'Nov'
   ELSE 'Dec'
 end as month_name , total_orders
from cte2
Order by total_orders
```

Row	customer_state	month_name	total_orders
1	SP	Aug	4982
2	SP	May	4632
3	SP	Jul	4381
4	SP	Jun	4104
5	SP	Mar	4047
6	SP	Apr	3967
7	SP	Feb	3357
8	SP	Jan	3351
9	SP	Nov	3012
10	SP	Dec	2357
11	SP	Oct	1908
12	SP	Sep	1648
13	RJ	May	1321
14	RJ	Aug	1307
15	RJ	Mar	1302

INSIGHTS : the most number of orders are placed in the state SP followed by RJ in the month of may , august .

- How are customers distributed across all states ?

```
select customer_state , count(distinct customer_unique_id) as
total_no_of_customer
from `target_sql_dsml.customers`
group by customer_state
```

Row	customer_state	total_no_of_custome
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
11	PE	1609
12	CE	1313
13	PA	949
14	MT	876
15	MA	726

INSIGHTS: we can clearly see that SP has the highest number of customers followed by RJ and MG which also tells me about the most no. of orders mystery.

4. **Impact on economy : Analyzing the money movement by ecommerce by looking at the order prices, freight and others**

- Percentage increase in the cost of orders from year 2017 to 2018

```
WITH
 cte1 AS (SELECT o.order_id,EXTRACT(YEAR FROM
o.order_purchase_timestamp) AS year, EXTRACT(MONTH FROM
o.order_purchase_timestamp) AS month,p.payment_value
 FROM `target_sql_dsml.orders` AS o
 JOIN `target_sql_dsml.payments` AS p
 ON o.order_id = p.order_id ), cte2 AS (
 SELECT *
 FROM cte1
 WHERE year BETWEEN 2017 AND 2018 AND month BETWEEN 1 AND 8 ),
```

```
 cte3 AS ( SELECT SUM(payment_value) AS total_value, year
 FROM cte2
 GROUP BY year )
SELECT year ,total_value , LAG(total_value) OVER (ORDER BY year)
AS previous_year_total, IFNULL(ROUND(total_value /
LAG(total_value) OVER (ORDER BY year), 2), 0)  AS rate
FROM cte3
order by year;
```

Row	year	total_value	previous_year_total	rate
1	2017	3669022.119999...	*null*	0.0
2	2018	8694733.840000...	3669022.119999...	2.37

INSIGHT: we can clearly see that the amount of orders has increased at the rate of 2.37 in 2018 from 2017 which means people bought more in this period might be because of their trust in the company and the quality level they managed to approve that satisfies most customers .

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

```
with cte1 as
(
 select c.customer_state ,o.order_id, p.payment_value
 from `target_sql_dsml.customers` as c
 join `target_sql_dsml.orders` as o
 on c.customer_id = o.customer_id
 join `target_sql_dsml.payments` as p
 on o.order_id = p.order_id
),cte2 as
(
```

```sql
 select customer_state , round(sum(payment_value),2) as
total_price , round((sum(payment_value)/count(distinct order_id))
, 2) as avg_price
 from cte1
 group by customer_state
)
select *
from cte2
order by cte2.total_price DESC, cte2.avg_price DESC;
```

Row	customer_state	total_price	avg_price
1	SP	5998226.96	143.69
2	RJ	2144379.69	166.85
3	MG	1872257.26	160.92
4	RS	890898.54	162.99
5	PR	811156.38	160.78
6	SC	623086.43	171.32
7	BA	616645.82	182.44
8	DF	355141.08	165.95
9	GO	350092.31	173.31
10	ES	325967.55	160.34
11	PE	324850.44	196.64
12	CE	279464.03	209.18
13	PA	218295.85	223.89
14	MT	187029.29	206.21
15	MA	152523.02	204.18

INSIGHTS : we can see that SP , RJ , MG are the top three states that order the most with highest payment value which can be due to the highest customer segment in these regions which we have seen earlier in previous analysis.

- Calculate the total and average value of order freight for each state

```
with cte1 as
(
 select c.customer_state ,o.order_id, oi.freight_value
 from `target_sql_dsml.customers` as c
 join `target_sql_dsml.orders` as o
 on c.customer_id = o.customer_id
 join `target_sql_dsml.order_items` as oi
 on o.order_id = oi.order_id
),cte2 as
(
 select customer_state , round(sum(freight_value),2) as
total_price , round((sum(freight_value)/count(distinct order_id))
, 2) as avg_price
 from cte1
 group by customer_state
)
select *
from cte2
order by total_price DESC , avg_price DESC;
```

Row	customer_state	total_price	avg_price
1	SP	718723.07	17.37
2	RJ	305589.31	23.95
3	MG	270853.46	23.46
4	RS	135522.74	24.95
5	PR	117851.68	23.58
6	BA	100156.68	29.83
7	SC	89660.26	24.82
8	PE	59449.66	36.07
9	GO	53114.98	26.46
10	DF	50625.5	23.82
11	ES	49764.6	24.58
12	CE	48351.59	36.44
13	PA	38699.3	39.9
14	MA	31523.77	42.6
15	MT	29715.43	32.91

INSIGHTS: SP , RJ , MG again tops the list , freight value may be affected due to weight of the item , distance where you have to send the package , company policy , mode of transportation like air , ship . These locations since they have the largest customer target base would eventually have high freight value and RJ is home to one of the brazil largest ports and these states are central to brazil economy with massive goods imports and exports.

5. Analysis based on sales , freight and delivery time

- Find the no. of days taken to deliver each order from the orders purchase date as delivery time , also calculate the the difference(in days)between the estimated and actual delivery date of an order

```
select order_id ,
date_diff(date(order_delivered_customer_date),date(order_purchase_
timestamp),day) as delivery_time ,
date_diff(date(order_estimated_delivery_date),date(order_delivered
_customer_date),day) as diff_estimated_delivery
from `target_sql_dsml.orders`
order by delivery_time DESC , diff_estimated_delivery DESC
```

Row	order_id	delivery_time	diff_estimated_deliv
1	ca07593549f1816d26a572e06...	210	-181
2	1b3190b2dfa9d789e1f14c05b...	208	-188
3	440d0d17af552815d15a9e41a...	196	-165
4	2fb597c2f772eca01b1f5c561b...	195	-155
5	285ab9426d6982034523a855f...	195	-166
6	0f4519c5f1c541ddec9f21b3bd...	194	-161
7	47b40429ed8cce3aee9199792...	191	-175
8	2fe324febf907e3ea3f2aa9650...	190	-167
9	2d7561026d542c8dbd8f0daea...	188	-159
10	c27815f7e3dd0b926b5855262...	188	-162
11	437222e3fd1b07396f1d9ba8c...	187	-144
12	dfe5f68118c2576143240b8d7...	186	-153
13	6e82dcfb5eada6283dba34f16...	183	-155
14	2ba1366baecad3c3536f27546...	181	-152
15	d24e8541128cea179a11a6517...	175	-161

INSIGHTS : we can clearly see that max delivery time taken in brazil to deliver is 210 days which was very late with -181 days might be due to remote place order which was very far from central location

- Find the top 5 states with highest and lowest average freight value

```
WITH cte1 AS (

 SELECT c.customer_state, ROUND(SUM(oi.freight_value) /
COUNT(DISTINCT o.order_id), 2) AS avg_freight_value

 FROM `target_sql_dsml.customers` AS c

 JOIN `target_sql_dsml.orders` AS o

 ON c.customer_id = o.customer_id

 JOIN `target_sql_dsml.order_items` AS oi

 ON o.order_id = oi.order_id

 GROUP BY c.customer_state

),

cte2 AS (

 SELECT customer_state, avg_freight_value

 FROM cte1

 ORDER BY avg_freight_value DESC

 LIMIT 5

),
```

```
cte3 AS (
 SELECT customer_state, avg_freight_value
 FROM cte1
 ORDER BY avg_freight_value ASC
 LIMIT 5
)
SELECT * FROM cte2
UNION ALL
SELECT * FROM cte3
ORDER BY avg_freight_value ASC;
```

Row	customer_state	avg_freight_value
1	SP	17.37
2	MG	23.46
3	PR	23.58
4	DF	23.82
5	RJ	23.95
6	PI	43.04
7	AC	45.52
8	RO	46.22
9	PB	48.35
10	RR	48.59

Note : row 1 to row 5 represents the top 5 states with lowest freight value in increasing order of their freight value. Row 5 to row 10 represents highest top 5 freight value in increasing order of their freight value.

INSIGHT : SP is the state of brazil with lowest freight value of 17.37 and RR is the state with highest avg freight value of about 48.49

- Find the top 5 states with highest and lowest average delivery time.

```
WITH cte1 AS (SELECT c.customer_state,
ROUND(SUM(DATE_DIFF(DATE(o.order_delivered_customer_date),
DATE(o.order_purchase_timestamp), DAY)) / COUNT(DISTINCT
o.order_id), 2) AS avg_delivery_time
 FROM `target_sql_dsml.customers` AS c
 JOIN `target_sql_dsml.orders` AS o
   ON c.customer_id = o.customer_id
 GROUP BY c.customer_state
),
cte2 AS (
 SELECT customer_state, avg_delivery_time
 FROM cte1
 ORDER BY avg_delivery_time DESC
 LIMIT 5
),
cte3 AS (
 SELECT customer_state, avg_delivery_time
 FROM cte1
 ORDER BY avg_delivery_time ASC
 LIMIT 5
)
```

```
SELECT * FROM cte2

UNION ALL

SELECT * FROM cte3

ORDER BY avg_delivery_time ASC;
```

Row	customer_state	avg_delivery_time
1	SP	8.44
2	PR	11.65
3	MG	11.66
4	DF	12.54
5	SC	14.54
6	PA	23.02
7	AL	23.55
8	AM	25.82
9	RR	26.15
10	AP	26.78

Note: first five row represents bottom 5 states with lowest delivery time with increasing order of value and bottom 5 represents top 5 states with highest delivery tie with increasing order of their value.

INSIGHTS : SP is the state with lowest delivery time and AP is the state with highest delivery time

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

```
WITH state_avg_dates AS (

 SELECT

   c.customer_state,
```

```
    AVG(TIMESTAMP_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY)) AS avg_delivery
  FROM `target_sql_dsml.customers` AS c
  JOIN `target_sql_dsml.orders` AS o
    ON c.customer_id = o.customer_id
  WHERE o.order_delivered_customer_date IS NOT NULL
  GROUP BY c.customer_state
)
SELECT
  customer_state,
  ROUND(avg_delivery, 2) AS avg_delivery
FROM state_avg_dates
ORDER BY avg_delivery DESC
LIMIT 5;
```

Row	customer_state	avg_delivery
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

INSIGHT : AC, RO , AP are among the top 3 with fastest delivery time .

6. Analysis based on payments

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

```
WITH cte AS (
 SELECT
   EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
   EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
   p.payment_type,
   COUNT(DISTINCT o.order_id) AS total_orders
 FROM `target_sql_dsml.orders` AS o
 JOIN `target_sql_dsml.payments` AS p
   ON o.order_id = p.order_id
 GROUP BY year, month, payment_type
)
SELECT *
FROM cte
ORDER BY year, month;
```

Row	year	month	payment_type	total_orders
1	2016	9	credit_card	3
2	2016	10	voucher	11
3	2016	10	credit_card	253
4	2016	10	UPI	63
5	2016	10	debit_card	2
6	2016	12	credit_card	1
7	2017	1	credit_card	582
8	2017	1	voucher	33
9	2017	1	UPI	197
10	2017	1	debit_card	9
11	2017	2	voucher	69
12	2017	2	credit_card	1347
13	2017	2	UPI	398
14	2017	2	debit_card	13
15	2017	3	voucher	123

- Find the no. of orders placed on the basis of the payment installments that have been paid.

```
SELECT p.payment_installments, COUNT(DISTINCT p.order_id) AS
total_orders

FROM `target_sql_dsml.payments` AS p

WHERE p.payment_value > 0 and p.payment_installments > 0

GROUP BY p.payment_installments

ORDER BY p.payment_installments;
```

Row	payment_installment	total_orders
1	1	49057
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623
8	8	4253
9	9	644
10	10	5315

INSIGHTS: there are more no. of orders of customers who have paid their first installments which is showing a reducing trend as we go downward the table which can means high abandonment rate after their first installments .

RECOMMENDATIONS:

1. Given the significant growth in order from 2016 to 2017 it is important to build on this early adoption phase. We can go for targeting marketing strategies by offering special discounts or exclusive product lines for repeated customers.
2. Since there is a notable increase in certain months we can go for seasonal campaign or offer discounts during key months such as during winter season and pre-christmas
3. Customers prefer to shop in the afternoon we can ensure that target stores are well-staffed during afternoon and we can also include happy hours or afternoon -specific discount and also highlight special deals during peak times
4. Since May, August shows high order volumes in SP and RJ , hence we can again go for targeted marketing in these regions like hosting local events , regional discounts during these months to cater to customer needs in these states.
5. Improve customer feedback systems , ensure quick response times to complaints and maintain high standards of product quality.
6. We can also partner with logistics companies to negotiate better shipping rates or introduce free shipping thresholds for states like SP, RJ, MG since they have the highest freight value; this could be an opportunity for us to increase sales by offering discounts.

7. We can focus on improving logistics infrastructure especially in remote areas by collaborating with third party logistics to speed up the last minute delivery.
8. For payments we can implement automated reminders about upcoming payments , introducing payment flexibility and offer incentives for early repayment discount or interest free EMI options .