Business Case 1

Description: Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analysing the given dataset to extract valuable insights and provide actionable recommendations.

What does the 'good' look like?

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

i) Data type of all columns in the "customers" table

Query:

```
SELECT
Column_name,
Data_type
FROM `target_dataset`.INFORMATION_SCHEMA.COLUMNS
WHERE table_name='customers'
```

Output

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

ii) Get the time range between which the orders were placed.

Query:

```
SELECT
MIN (order_purchase_timestamp) as Start_time,
MAX (order_purchase_timestamp) as End_time
FROM`target_dataset.orders`
```

Output

Row	Start_time	End_time
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

- The above query gives the time range for the entire data set to get the time of order placed for each date we give the following query

QUERY:

```
SELECT
date_opt,
min(time_opt) as start_time,
max(time_opt) as end_time,
FROM
(
SELECT
order_purchase_timestamp,
Extract(date from order_purchase_timestamp ) as date_opt,
Extract(time from order_purchase_timestamp AT TIME ZONE "UTC") as
time_opt
from `target_dataset.orders`
)
group by date_opt
```

OUTPUT

Row	date_opt	start_time	end_time
1	2017-11-25	00:04:31	23:59:30
2	2017-12-05	00:05:32	23:58:33
3	2018-02-09	00:04:33	23:56:01
4	2017-11-06	00:05:50	23:40:20
5	2017-04-20	00:28:39	23:13:48
6	2017-07-13	00:06:19	23:35:55
7	2017-07-11	00:12:48	23:44:38
8	2017-07-29	00:23:27	23:42:30
9	2017-07-19	00:12:19	23:51:41
10	2018-05-11	00:09:19	23:49:18

Insight: When we look at the output and observe the start time and end time we can see that on most of the days approximately the order purchase span of customers is 24 hours.

iii) Count the number of Cities and States in our dataset

QUERY:

```
SELECT
COUNT (distinct geolocation city) as No_of_city,
COUNT (distinct geolocation_state) as No_of_state
FROM `target_dataset.geolocation`
```

Output

Row	No_of_city	No_of_state
1	8011	27

- We also take the count of city and state according to the customers

QUERY:

```
SELECT
count(distinct customer_city) as No_of_city,
count(distinct customer_state) as No_of_state
FROM `target_dataset.customers`
```

OUTPUT:

Row	No_of_city	No_of_state
1	4119	27

Insight: The count of city from customers table is lesser than the count in geolocations, so for increasing the reach of the company some steps could be taken.

QUESTION 2: In-depth Exploration:

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

QUERY:

```
SELECT
year,
month,
COUNT (order_id) as No_of_orders
FROM
(
SELECT
order_id,
order_purchase_timestamp,
Extract (year from order_purchase_timestamp ) as year,
Extract (month from order_purchase_timestamp ) as month
from `target_dataset.orders`
)
GROUP BY year,month
ORDER BY year, month
```

Output:

year	month	No_of_orders
2016	9	4
2016	10	324
2016	12	1
2017	1	800
2017	2	1780
2017	3	2682
2017	4	2404
2017	5	3700
2017	6	3245
2017	7	4026
2017	8	4331
2017	9	4285
2017	10	4631
2017	11	7544
2017	12	5673
2018	1	7269
2018	2	6728
2018	3	7211
2018	4	6939
2018	5	6873
2018	6	6167
2018	7	6292
2018	8	6512
2018	9	16
2018	10	4

- When we just look at the look at the year wise change in the sales we use the following query and take the given below table as the output

```
SELECT
year,
COUNT(order_id) as No_of_orders
FROM
(
SELECT
order_id,
order_purchase_timestamp,
Extract(year from order_purchase_timestamp ) as year,
from `target_dataset.orders`
)
GROUP BY year
ORDER BY year
```

OUTPUT:

Row	year	No_of_orders
1	2016	329
2	2017	45101
3	2018	54011

Insight: There is an overall increase in the no of orders over the years and approximately 10k new orders are placed in the year 2018 but this is not on continuous spectrum. Winters of 2017 has more orders placed compared to that of 2018

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

QUERY:

```
SELECT
year,
COUNT (order_id) as No_of_orders
FROM
(
SELECT
order_id,
order_purchase_timestamp,
Extract (month from order_purchase_timestamp ) as month
from `target_dataset.orders`
)
GROUP BY year
ORDER BY year
```

OUTPUT:

Row	month	No_of_orders
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573

Row	month	No_of_orders
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544
12	12	5674

Insight: There is a seasonality present in the patterns of order being placed. The maximum numbers of orders are being placed from the month of May to August. There is dip in the orders after September and again at increases slowly after October.

iii) 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

QUERY:

```
SELECT

CASE WHEN (time_hour between 0 and 6)
then ("Dawn")
WHEN (time_hour between 7 and 12)
then ("Mornings")
WHEN (time_hour between 13 and 18)
then ("Afternoon")
else ("Night")
end as Shifts,
count (order_id) as no_of_orders
FROM
(
SELECT
order_id,
order_purchase_timestamp,
Extract (hour from order_purchase_timestamp) as time_hour
FROM `target_dataset.orders`

order by order_purchase_timestamp desc, time_hour
)
group by Shifts
ORDER BY no_of_orders DESC
```

OUTPUT:

Row	Shifts	no_of_orders
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

Insight: Afternoon has the maximum no of orders followed by Night and least is in the dawn

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

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

QUERY:

```
SELECT
c.customer_state,
Extract (year from order_purchase_timestamp ) as year,
Extract(month from o.order_purchase_timestamp) as t_month,
count(o.order_id) as no_of_orders
FROM `target_dataset.customers` as c
left join `target_dataset.orders` as o
on c.customer_id= o.customer_id
group by year,c.customer_state, t_month
```

OUTPUT:

Row	customer_state	year	t_month	no_of_orders
4	CE	2018	2	88
5	CE	2018	3	98
6	CE	2017	5	62

Row	customer_state	year	t_month	no_of_orders
7	CE	2017	4	43
8	CE	2018	5	74
9	RS	2018	3	418
10	RS	2018	6	305
11	SC	2017	8	159
12	SC	2017	12	193

Insight: Different states on the same month has different pattern of orders, so to maximize the sales and expand the business strategy has to be different locally.

ii) How are the customers distributed across all the states?

QUERY

```
SELECT
customer_state,
Count(customer_id) No_of_Customers
FROM `target_dataset.customers`
GROUP BY customer_state
```

OUTPUT:

Row	customer_state	No_of_Customers
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747
11	PE	1652

Insight: The customer distribution across different states tells us about ours business approach locally.

QUESTION4: Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

QUERY:

```
SELECT*,
round (((cost_of_orders-prev_year_cost)/prev_year_cost)*100,2) as
percentage_increase
FROM
(
SELECT *,
lag (cost_of_orders,1) over (order by o_year) as prev_year_cost
FROM
(
SELECT
o_year,
sum (payment_value) as cost_of_orders
FROM
(
SELECT
Extract (year from od.order_purchase_timestamp ) as o_year,
Extract (month from od.order_purchase_timestamp) as t_month,
payment_value
FROM `target_dataset.orders` as od
left join `target_dataset.payments` as py
on od.order_id=py.order_id
)
where t_month BETWEEN 01 and 08
Group by o_year
HAVING o_year =2017 OR o_year=2018
)
)
ORDER BY o_year DESC
limit 1
```

OUTPUT:

Row	o_year	cost_of_orders	prev_year_cost	percentage_increase
1	2018	8694733.84	3669022.12	136.98

Insight: The cost of orders gave more than 100 % growth from the year 2017 to 2018.

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

QUERY:

```
SELECT
c.customer_state,
round (sum (odi.price), 2) as Total_order_price,
round (avg (odi.price), 2) as avg_order_price


FROM `target_dataset.order_items` as odi
join `target_dataset.orders` as o
on odi.order_id=o.order_id
join `target_dataset.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state
order by c.customer_state
```

OUTPUT:

Row	customer_state	Total_order_price	avg_order_price
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2

Insight: BA has the highest total order price although the average order price is not that high (134.6), which shows that number of orders placed in BA are quite significant

iii) Calculate the Total & Average value of order freight for each state

QUERY:

```
SELECT
c.customer_state,
round(sum(odi.freight_value),2) as Total_order_freight,
round(avg(odi.freight_value),2) as avg_order_freight


FROM `target_dataset.order_items` as odi
join `target_dataset.orders` as o
on odi.order_id=o.order_id
join `target_dataset.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state
order by c.customer_state
```

OUTPUT:

Row	customer_state	Total_order_freight	avg_order_freight
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77
10	MA	31523.77	38.26

Insight: The average Order freight price significantly shows that it is one of the major reasons for BA having highest total order price.

Question 5) Analysis based on sales, freight and delivery time.

i) **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 & actual delivery date of an order.
Do this in a single query.

QUERY:

```
SELECT
order_id,
order_purchase_timestamp as opt,
time_taken_to_delivery as ttd,
diff_estimated_delivery as d_e_d
FROM
(
SELECT
order_id,
order_purchase_timestamp ,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)
as time_taken_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,
day) as diff_estimated_delivery
FROM `target_dataset.orders`
where order_status = 'delivered'
)
```

OUTPUT:

Row	order_id	opt	ttd	d_e_d
1	635c894d068ac37e6e03dc54eccb6189	2017-04-15 15:37:38 UTC	30	1
2	3b97562c3aee8bdedcb5c2e45a50d5e1	2017-04-14 22:21:54 UTC	32	0
3	68f47f50f04c4cb6774570cfde3a9aa7	2017-04-16 14:56:13 UTC	29	1
4	276e9ec344d3bf029ff83a161c6b3ce9	2017-04-08 21:20:24 UTC	43	-4
5	54e1a3c2b97fb0809da548a59f64c813	2017-04-11 19:49:45 UTC	40	-4
6	fd04fa4105ee8045f6a0139ca5b49f27	2017-04-12 12:17:08 UTC	37	-1
7	302bb8109d097a9fc6e9cefc5917d1f3	2017-04-19 22:52:59 UTC	33	-5
8	66057d37308e787052a32828cd007e58	2017-04-15 19:22:06 UTC	38	-6
9	19135c945c554eebfd7576c733d5ebdd	2017-07-11 14:09:37 UTC	36	-2
10	4493e45e7ca1084efcd38ddebf174dda	2017-07-11 20:56:34 UTC	34	0

Insight: The bigger positive value of d_e_d column shows fast delivery of the order, smaller the number slower the delivery from the estimated time.

ii) Find out top 5 states with and lowest average freight value

QUERY:

```
SELECT *
FROM
(
SELECT
c.customer_state,
round(avg(odi.freight_value),2) as avg_order_freight,
'top 5' as avg_order_type
FROM `target_dataset.order_items` as odi
join `target_dataset.orders` as o
on odi.order_id=o.order_id
join `target_dataset.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state
order by avg_order_freight desc
limit 5
)
UNION ALL
(
SELECT
c.customer_state,
-- round(sum(odi.freight_value),2) as Total_order_freight,
round(avg(odi.freight_value),2) as avg_order_freight,
"bottom 5" as avg_order_type
FROM `target_dataset.order_items` as odi
join `target_dataset.orders` as o
on odi.order_id=o.order_id
join `target_dataset.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state
order by avg_order_freight
 limit 5
)
```

OUTPUT:

Row	customer_state	avg_order_freight	avg_order_type
1	RR	42.98	top 5
2	PB	42.72	top 5
3	RO	41.07	top 5
4	AC	40.07	top 5
5	PI	39.15	top 5
6	SP	15.15	bottom 5
7	PR	20.53	bottom 5
8	MG	20.63	bottom 5
9	RJ	20.96	bottom 5
10	DF	21.04	bottom 5

iii) Find out the top 5 states with highest and lowest average delivery time

QUERY:

```
SELECT*
FROM
(
SELECT
c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_times
tamp,day)),0) as delivery_time,
'top 5' as sorted_by
FROM `target_dataset.orders` as o
join `target_dataset.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state
order by delivery_time desc
limit 5
)
UNION ALL
(
SELECT
c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_times
tamp,day)),0) as delivery_time,
'bottom 5' as sorted_by
FROM `target_dataset.orders` as o
join `target_dataset.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state
order by delivery_time
limit 5
)
```

OUTPUT:

Row	customer_state	delivery_time	sorted_by
1	RR	29.0	top 5
2	AP	27.0	top 5
3	AM	26.0	top 5
4	AL	24.0	top 5
5	PA	23.0	top 5
6	SP	8.0	bottom 5
7	MG	12.0	bottom 5
8	PR	12.0	bottom 5
9	DF	13.0	bottom 5
10	SC	14.0	bottom 5

Insight: The delivery time and freight price are correlated in some way and this can be observed by looking at previous two tables

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

QUERY:

```
SELECT
c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_estimated_deli
very_date,day)),0) as Quickness_parameter,
FROM `target_dataset.orders` as o
join `target_dataset.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state
Order by Quickness_parameter
LIMIT 5
```

OUTPUT:

Row	customer_state	Quickness_parameter
1	AC	-20.0
2	RO	-19.0
3	AM	-19.0
4	AP	-19.0
5	RR	-16.0

Insight: RR, AP, AM although being in top 5 in the delivery time yet are amount the fastest ones because the delivery is being delivered as promised and even way before

QUESTON 6: Analysis based on the payments:

i) **Find the month on month no. of orders placed using different payment types**.

QUERY:

```
SELECT
py.payment_type as Payment_Type,
Extract(year from o.order_purchase_timestamp ) as o_year,
```

```
Extract(month from o.order_purchase_timestamp) as t_month,
count(o.order_id) as No_of_orders
FROM `target_dataset.orders` as o
join `target_dataset.payments` as py
on o.order_id = py.order_id
group by o_year,t_month, py.payment_type
```

OUTPUT:

Row	Payment_Type	o_year	t_month	No_of_orders
1	UPI	2017	11	1509
2	credit_card	2017	12	4377
3	UPI	2018	2	1325
4	credit_card	2017	11	5897
5	voucher	2017	4	202
6	credit_card	2017	7	3086
7	UPI	2017	7	845
8	credit_card	2018	5	5497
9	credit_card	2017	10	3524
10	credit_card	2018	1	5520

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

QUERY:

```
SELECT
 payment_sequential,
 payment_installments,
 count(order_id) as count_order
FROM  `target_dataset.payments`
WHERE payment_installments >= payment_sequential
GROUP BY payment_sequential, payment_installments
```

OUTPUT:

Row	payment_sequential	payment_installments	count_order
1	1	1	48236
2	1	2	12360
3	2	2	53
4	1	3	10422
5	2	3	38

Row	payment_sequential	payment_installments	count_order
6	3	3	1
7	1	4	7066
8	2	4	32
9	1	5	5221
10	2	5	18

Recommendations

- To have more number of cities included in the active customers list in order to have bigger crowd to serve, this difference was observed in part iii) of the first question
- Giving better offers like more discounts and buy few get 1 free type offers in the months where there is bigger dip in sales i.e. in the offseason to attract more customers
- Having more inventory and better delivery systems to avoid delay in the delivery time to avoid loss of customers over the years
- Customers with good purchase history has to be offered special privileges in terms of special discounts to retain them for longer timespan
- Average purchasing power of customers has to be kept in mind while having stores with costly products in those states where sales are low