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

1. Data type of columns in a table

ANS: If we see the table Payments,

a. the first column is order_id which is of VARCHAR type containing the order id from 2016-2018(if we join it with orders table which contains the order id between 206-2018).
b. The second col is payment_sequence which is int type. Third is payment_type which contains values like credit_card,voucher,not_defined,debit_card.
c. 4^{t} column is payment_installment which is also int type .
d. 5th column is payment_value which is float type.

2. Time period for which the data is given

ANS: For orders dataset the data is given from 2016-2018

```
7  SELECT
8  MIN(order_purchase_timestamp),
9  MAX(order_purchase_timestamp)
0  from `akash-218509.TargetDataset.orders`
```

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETA
v	f0_	f1_	
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	

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

ANS:

```
12
13
14  SELECT
15  DISTINCT
16  c.customer_id,
17  c.customer_city,
18  c.customer_state
19  from `akash-218509.TargetDataset.orders` as o join `akash-218509.TargetDataset.customers` as c on
20  o.customer_id=c.customer_id
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

ow	customer_id	customer_city	customer_state
1	0735e7e4298a2ebbb4664934...	acu	RN
2	903b3d86e3990db01619a4eb...	acu	RN
3	38c97666e962d4fea7fd6a83e...	acu	RN
4	77c2f46cf580f4874c9a5751c2...	ico	CE
5	4d3ef4cfffb8ad4767c199c36a...	ico	CE
6	3000841b86e1fbe9493b52324...	ico	CE
7	3c325415ccc7e622c66dec4bc...	ico	CE
8	04f3a7b250e3be964f01bf22bc...	ico	CE
9	894202b8ef01f4719a4691e79...	ico	CE

2. In-depth Exploration:

1. 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
MONTH,YEAR,
COUNT(*) as OrderPurchasedCount
FROM(
SELECT
  order_id,
  order_purchase_timestamp,
  BrazilDateTimeZone,
  EXTRACT(DAY FROM BrazilDateTimeZone) as DAY,
  EXTRACT(MONTH FROM BrazilDateTimeZone) as MONTH,
  EXTRACT(YEAR FROM BrazilDateTimeZone) as YEAR,
  EXTRACT(DAYOFWEEK FROM BrazilDateTimeZone) as DAYOFWEEK,
  CAST(BrazilDateTimeZone AS TIME) as brazilTime
    FROM(
```

```
        SELECT
        order_id,
        order_purchase_timestamp,
        DATETIME(order_purchase_timestamp,"Brazil/West") as Brazi
lDateTimeZone
        FROM `akash-218509.TargetDataset.orders`))
group by  MONTH,YEAR order by YEAR,MONTH
```

ow	MONTH	YEAR	OrderPurchased
1	9	2016	4
2	10	2016	324
3	12	2016	1
4	1	2017	801
5	2	2017	1783
6	3	2017	2681
7	4	2017	2402
8	5	2017	3707
9	6	2017	3241
10	7	2017	4028
11	8	2017	4337
12	9	2017	4280
13	10	2017	4630
14	11	2017	7554

Insights:There is sudden spike of orders for the month of Nov 2017,looking further on daily trend on 24th Nov 2017 there was Black Friday Sale
https://www.independent.co.uk/news/world/black-friday-2017-brazil-shoppers-discount-sales-brazil-south-africa-a8073651.html

Recommendation:During the sale time inventory should should be full.Based on customer analysis and discounts on specific products their inventory should be specially taken care for.

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

```
  SELECT
  DayPart,
  COUNT(DayPart) As DayPartCount
  FROM(
```

```
  select
    order_purchase_timestamp,
    BrazilDateTimeZone,
    brazilTime,
    case
      when brazilTime between cast('00:00:00' as time) and cast('
05:59:59' as time) then "Dawn"
      when brazilTime between cast('06:00:00' as time) and cast('
11:59:59' as time) then "Morning"
      when brazilTime between cast('12:00:00' as time) and cast('
17:59:59' as time) then "Afternoon"
      when brazilTime between cast('18:00:00' as time) and cast('
23:59:59' as time) then "Night"
    END as DayPart
    FROM(
        SELECT
        order_purchase_timestamp,
        BrazilDateTimeZone,
        CAST(BrazilDateTimeZone AS TIME) as brazilTime
          FROM(
              SELECT
              order_purchase_timestamp,
              DATETIME(order_purchase_timestamp,"Brazil/West") as
 BrazilDateTimeZone
              FROM `akash-218509.TargetDataset.orders`)))
  group by DayPart
```

Row	DayPart	DayPartCount
1	Morning	38291
2	Night	14285
3	Afternoon	36986
4	Dawn	9879

Insights: Customer tends to buy most during the morning and afternoon time.
Recommendation: special offer,discount notifications should be sent

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

1. Get month on month orders by states

```sql
WITH orderBrazil AS(
SELECT
  order_id,
  customer_id,
  order_purchase_timestamp,
  BrazilDateTimeZone,
  EXTRACT(DAY FROM BrazilDateTimeZone) as DAY,
  EXTRACT(MONTH FROM BrazilDateTimeZone) as MONTH,
  EXTRACT(YEAR FROM BrazilDateTimeZone) as YEAR,
  EXTRACT(DAYOFWEEK FROM BrazilDateTimeZone) as DAYOFWEEK,
  CAST(BrazilDateTimeZone AS TIME) as brazilTime
  FROM(
SELECT
        order_id,
        customer_id,
        order_purchase_timestamp,
        DATETIME(order_purchase_timestamp,"Brazil/West") as Brazi
lDateTimeZone
        FROM `akash-218509.TargetDataset.orders`))
SELECT
MONTH,YEAR,customer_state,
COUNT(*) as OrderCount
 FROM orderBrazil as ob
join `akash-218509.TargetDataset.customers` as c
on ob.customer_id=c.customer_id
group by MONTH,YEAR,customer_state
order by YEAR,MONTH
```

ow	MONTH	YEAR	customer_state	OrderCount
1	9	2016	RR	1
2	9	2016	RS	1
3	9	2016	SP	2
4	10	2016	SP	113
5	10	2016	RS	24
6	10	2016	BA	4
7	10	2016	PR	19
8	10	2016	RJ	56
9	10	2016	RN	4
10	10	2016	MT	3
11	10	2016	PB	1
12	10	2016	SC	11
13	10	2016	AL	2
14	10	2016	PE	7

2. Distribution of customers across the states in Brazil

```
SELECT
COUNT(c.customer_id) as Customer_count,
c.customer_state
FROM `akash-218509.TargetDataset.customers` as c join
`akash-218509.TargetDataset.orders` as o
on c.customer_id=o.customer_id
group by c.customer_state
order by Customer_count desc
```

Row	Customer_count	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF
9	2033	ES
10	2020	GO
11	1652	PE
12	1336	CE
13	975	PA
14	907	MT

Insights: TOP 3 states SP,RJ and MG are customer base of about 2/3 of the total customers

Recommendation:Inventory should always be up to the mark for these high customer_count states.

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

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

```
SELECT
customer_state,
AVG(oi.price+oi.freight_value) as mean,
SUM(oi.price+oi.freight_value) as Sum
FROM `akash-218509.TargetDataset.customers` as c join `akash-218509.TargetDataset.orders` as o
on c.customer_id=o.customer_id join `akash-218509.TargetDataset.order_items` as oi on oi.order_id=o.order_id
group by customer_state
order by mean
```

Row	customer_state	mean	Sum
1	SP	124.800904...	5921678.11...
2	PR	139.535790...	800935.439...
3	MG	141.378740...	1856161.49...
4	RS	142.073257...	885826.759...
5	ES	143.972477...	324801.910...
6	RJ	146.078742...	2129681.98...
7	SC	146.123946...	610213.599...
8	DF	146.811903...	353229.440...
9	GO	149.038546...	347706.930...
10	BA	160.965167...	611506.670...
11	MS	166.003260...	135956.669...
12	AM	168.701393...	27835.7299...
13	MT	176.463469...	186168.960...
14	PE	178.426184...	322237.689...

Insight: The mean freight value for states like SP,PR are very high and their estimated delivery time is also very high
Recommendation: Use of Delivery Partner/Warehouse near these states

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```
select * from(
select
```

```
order_id,
DATE_DIFF(order_estimated_delivery_date,order_purchase_times
tamp,DAY) as EstimatedDays,
DATE_DIFF(order_delivered_customer_date,order_purchase_times
tamp,DAY) as ActualDays
from `akash-218509.TargetDataset.orders`)
WHERE EstimatedDays is NOT NULL and ActualDays is Not NULL
```

v	order_id	EstimatedDays	ActualDays
1	635c894d068ac37e6e03dc54e...	32	30
2	3b97562c3aee8bdedcb5c2e45...	33	32
3	68f47f50f04c4cb6774570cfde...	31	29
4	276e9ec344d3bf029ff83a161c...	39	43
5	54e1a3c2b97fb0809da548a59...	36	40
6	fd04fa4105ee8045f6a0139ca5...	35	37
7	302bb8109d097a9fc6e9cefc5...	28	33
8	66057d37308e787052a32828...	32	38
9	19135c945c554eebfd7576c73...	33	36
10	4493e45e7ca1084efcd38ddeb...	33	34
11	70c77e51e0f179d75a64a6141...	31	42
12	d7918e406132d7c81f1b84527...	31	35
13	43f6604e77ce6433e7d68dd86...	25	32
14	37073d851c3f30deebe598e5a...	22	31

Insights: For most of the cases the numbers are close to actuals.

Recommendation:

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

time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

```
select * from
(
select
o.order_id,
DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_t
imestamp,DAY) as time_to_delivery,
```

```
DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_
delivery_date,DAY) as diff_estimated_delivery
from `akash-218509.TargetDataset.orders` as o)
where time_to_delivery is NOT NULL AND diff_estimated_delive
ry is NOT NULL
```

Row	order_id	time_to_delivery	diff_estimated_c
1	1950d777989f6a877539f5379...	30	12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	-28
3	65d1e226dfaeb8cdc42f66542...	35	-16
4	635c894d068ac37e6e03dc54e...	30	-1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	-1
7	276e9ec344d3bf029ff83a161c...	43	4
8	54e1a3c2b97fb0809da548a59...	40	4
9	fd04fa4105ee8045f6a0139ca5...	37	1
10	302bb8109d097a9fc6e9cefc5...	33	5
11	66057d37308e787052a32828...	38	6
12	19135c945c554eebfd7576c73...	36	2
13	4493e45e7ca1084efcd38ddeb...	34	0
14	70c77e51e0f179d75a64a6141...	42	11

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

```
select
c.customer_state,
AVG(oi.freight_value) as MeanFreightValue,
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purcha
se_timestamp,DAY)) as time_to_delivery,
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_estima
ted_delivery_date,DAY)) as diff_estimated_delivery
from `akash-218509.TargetDataset.customers` as c join `akash
-218509.TargetDataset.orders` as o
 on c.customer_id = o.customer_id
join `akash-218509.TargetDataset.order_items` oi
 on oi.order_id=o.order_id
 group by c.customer_state
 order by MeanFreightValue desc
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION

Row	customer_state	MeanFreightVal	time_to_delivery	diff_estimated_c
1	RR	42.9844230...	27.8260869...	-17.434782...
2	PB	42.7238039...	20.1194539...	-12.150170...
3	RO	41.0697122...	19.2820512...	-19.080586...
4	AC	40.0733695...	20.3296703...	-20.010989...
5	PI	39.1479704...	18.9311663...	-10.682600...
6	MA	38.2570024...	21.2037500...	-9.1099999...
7	TO	37.2466031...	17.0032258...	-11.461290...
8	SE	36.6531688...	20.9786666...	-9.1653333...
9	AL	35.8436711...	23.9929742...	-7.9765807...
10	PA	35.8326851...	23.3017077...	-13.374762...
11	RN	35.6523629...	18.8733205...	-13.055662...
12	AP	34.0060975...	27.7530864...	-17.444444...
13	AM	33.2053939...	25.9631901...	-18.975460...
14	PE	32.9178626...	17.7920962...	-12.552119...

Insights:For states where average time_to_delivery is higher the freight value itself is higher.

Recommendation: Warehouses around these states could bring down freight charges signifcantly

4.5 Sort the data to get the following:

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

```
select
c.customer_state as customer_state_AVGhigh_freight,
AVG(oi.freight_value) as Average_freight_value
from `akash-218509.TargetDataset.orders` as o join `akash-21
8509.TargetDataset.order_items` as oi
on o.order_id=oi.order_id join `akash-218509.TargetDataset.c
ustomers` as c on c.customer_id=o.customer_id
group by c.customer_state
order by Average_freight_value desc
limit 5
```

JOB INFORMATION | RESULTS | JSON | EXE

Row	customer_state_AVGhigh_freight	Average_freight_
1	RR	42.9844230...
2	PB	42.7238039...
3	RO	41.0697122...
4	AC	40.0733695...
5	PI	39.1479704...

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

```
select
c.customer_state as customer_state_AVGlow_freight,
AVG(oi.freight_value) as Average_freight_value
from `akash-218509.TargetDataset.orders` as o join `akash-21
8509.TargetDataset.order_items` as oi
on o.order_id=oi.order_id join `akash-218509.TargetDataset.c
ustomers` as c on c.customer_id=o.customer_id
group by c.customer_state
order by Average_freight_value asc limit 5
```

tow	customer_state_AVGlow_freight	Average_freight_
1	SP	15.1472753...
2	PR	20.5316515...
3	MG	20.6301668...
4	RJ	20.9609239...
5	DF	21.0413549...

INSIGHTS: For states where delivery time is less has the least freight value and for states where the delivery time is high has the most freight value

Recommendation: Setup of warehouses near top states with high reight value

4.6. Top 5 states with highest average time to delivery

```sql
select
c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_
purchase_timestamp,DAY))) as TimeForDelivery
from `akash-218509.TargetDataset.customers` as c join `akash
-218509.TargetDataset.orders` as o
on c.customer_id=o.customer_id
group by c.customer_state
order by TimeForDelivery desc
limit 5
```

JOB INFORMATION | RESULTS | JSON | EXECU

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

Top 5 states with lowest average time to delivery

```sql
select
c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_
purchase_timestamp,DAY))) as TimeForDelivery
from `akash-218509.TargetDataset.customers` as c join `akash
-218509.TargetDataset.orders` as o
on c.customer_id=o.customer_id
group by c.customer_state
order by TimeForDelivery asc
limit 5
```

tow	customer_state	TimeForDelivery
1	SP	8.0
2	MG	12.0
3	PR	12.0
4	DF	13.0
5	SC	14.0

Insights: Few states average delivery time is around a month

Recommendation: Use of delivery partner for those states where Target has difficulty/takes more time to deliver.

4.7 Top 5 states where delivery is really fast compared to estimated date

```
select
c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date,o.order_
delivered_customer_date,DAY))) as avg_delivery_time
from `akash-218509.TargetDataset.customers` as c join `akash
-218509.TargetDataset.orders` as o
on c.customer_id=o.customer_id
group by c.customer_state
order by avg_delivery_time asc
limit 5
```

w	customer_state	avg_delivery_tim
1	AL	8.0
2	MA	9.0
3	SE	9.0
4	ES	10.0
5	SP	10.0

Top 5 states where delivery is not so fast compared to estimated date

```
select
c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_
estimated_delivery_date,DAY))) as avg_delivery_time
from `akash-218509.TargetDataset.customers` as c join `akash
-218509.TargetDataset.orders` as o
on c.customer_id=o.customer_id
group by c.customer_state
order by avg_delivery_time asc
limit 5
```

JOB INFORMATION | RESULTS | JSON | EXEC

w	customer_state	avg_delivery_tim
1	AC	-20.0
2	RO	-19.0
3	AM	-19.0
4	AP	-19.0
5	RR	-16.0

Insights:For Few states the delivery date is more than 15 days from the estimated date.

Recommendation: Setup of warehouse/delivery partner for these states.

6. Payment type analysis:

1. Month over Month count of orders for different payment types

```
WITH orderBrazil AS(
       SELECT
         order_id,
         order_purchase_timestamp,
         BrazilDateTimeZone,
         EXTRACT(MONTH FROM BrazilDateTimeZone) as MONTH,
         EXTRACT(YEAR FROM BrazilDateTimeZone) as YEAR
         FROM(
       SELECT
               order_id,
               customer_id,
               order_purchase_timestamp,
               DATETIME(order_purchase_timestamp,"Brazil/West") as BrazilDateTi
       meZone
               FROM `akash-218509.TargetDataset.orders`))
       SELECT
       p.payment_type,
       ob.MONTH,ob.YEAR,
       COUNT(p.payment_type) as payment_type_count
        FROM orderBrazil as ob
       join `akash-218509.TargetDataset.payments` as p
       on ob.order_id=p.order_id
       group by MONTH,YEAR,payment_type
       order by YEAR,MONTH
```

Row	payment_type	MONTH	YEAR	payment_type_c
1	credit_card	9	2016	3
2	credit_card	10	2016	254
3	voucher	10	2016	23
4	debit_card	10	2016	2
5	UPI	10	2016	63
6	credit_card	12	2016	1
7	voucher	1	2017	61
8	UPI	1	2017	197
9	credit_card	1	2017	584
10	debit_card	1	2017	9
11	credit_card	2	2017	1356

Insights: credit card is the most used payment method.

Recommendation: Points or offers on payment on particular credit cards or in general credit cards

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

```
       select
       p.payment_installments,
```

```
COUNT(p.order_id) as OrderIdCount
 from `akash-218509.TargetDataset.payments` as p join `akash-218509.Targ
etDataset.orders` as o on p.order_id=o.order_id
 group by payment_installments
```

Row	payment_installı	OrderIdCount
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

Insights: Majority of people tends to pay out in 1 installment, but there is significant amount of people which tends to pay till 10 installments.

Recommendation: NO cost EMI for people paying under an year