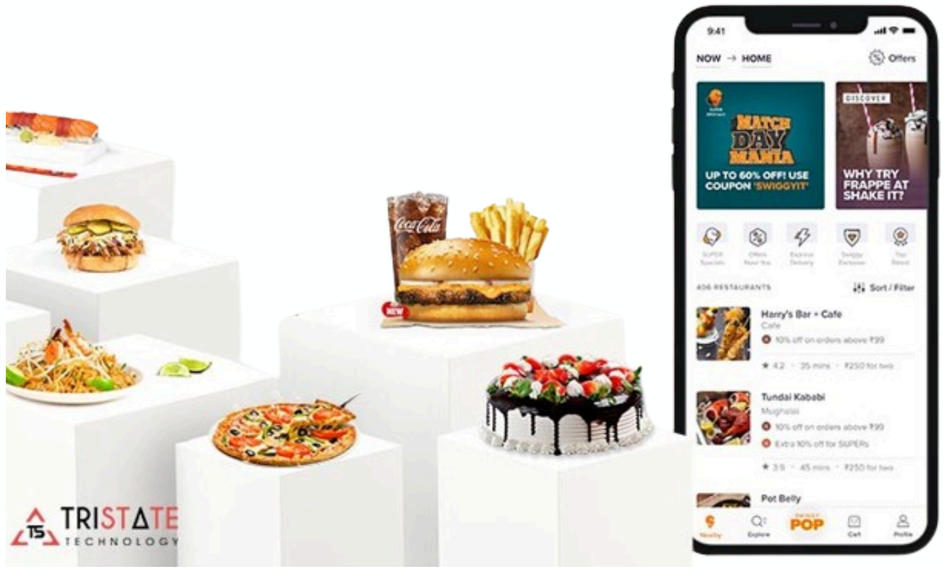
Swiggy Data Analysis

AQSA JABEEN

Table of content

Project Introduction

Overview



Swiggy play a pivotal role in connecting customers with a wide variety of restaurants. As competition intensifies, it becomes increasingly crucial for such platforms to leverage data to enhance their service offerings, optimize operations, and improve customer satisfaction. This project aims to harness the wealth of data available from Swiggy's operations to extract meaningful insights that can drive informed business decisions, improve customer loyalty, and ultimately contribute to the platform's success.

Expected Outcomes

- Enhance customer satisfaction and loyalty by better understanding and meeting customer needs.
- Support restaurants in optimizing their operations and offerings.
- Improve overall platform performance through data-driven decision-making.
- Foster a competitive edge in the dynamic food delivery market.

Methadology

Data Collection

- **Sources:** Swiggy's databas**es.**
- **Tables:** users, restaurants, food, menu, orders, order_details, delivery_partners.

Data Preparation

Cleaning: Handle missing values, remove duplicates, ensure consistency.
Transformation: Convert data types, extract date components (e.g., month, year).

Data Integration

- Integrate **order**s with **users**, **restaurants,** and **order_details**.
- Join **order_details** with **food**.
- Join **orders** with **delivery_partners.**

Finding #1

1. **Find customers who have never ordered?**

Query:

```
SELECT
   u.user_id, u.name
FROM
   users u
      LEFT JOIN
   orders o ON u.user_id = o.user_id
WHERE
   o.order_id IS NULL;
```

Result

Result Grid | Filter Rows:

	user_id	name
▶	6	Anupama
	7	Rishabh

Result 9

Finding #2

1. **Average Price/dish**

Query:

```
SELECT
    fd.f_name, ROUND(AVG(m.price)
AS Average_price
FROM
  food fd
     INNER JOIN
  menu m ON fd.f_id - m.f_id
GROUP BY fd.f_name;
```

Result

Result Grid | Filter Rows:

f_name	Average_price
Veg Manchurian	212.50
Schezwan Noodles	209.64
Rava Idli	216.79
Masala Dosa	212.50
Roti meal	215.36
Rice Meal	209.58
Chicken Popcorn	203.93

Finding #3

Find the top restaurant in terms of the number of orders for a given month

Query:

```
SELECT
   *
FROM
   orders;
SELECT
   rt.r_name, COUNT(o.order_id)
FROM
   orders o
      INNER JOIN
   restaurants rt ON o.r_id = rt.r_id
WHERE
   MONTH(o.date) = 06
GROUP BY rt.r_name
ORDER BY rt.r_name DESC
LIMIT 1;
```

Result

Result Grid Filter Rows:

f_name	Average_price
Veg Manchurian	212.50
Schezwan Noodles	209.64
Rava Idli	216.79
Masala Dosa	212.50
Roti meal	215.36
Rice Meal	209.58
Chicken Popcorn	203.93

Finding #4

1. **Restaurants with monthly sales greater than x for**

Query:

```
-- let x=500
with resturant_monthly_sale as
(
select rt.r_name, sum(o.amount) as
Total_sale,
date_format(o.date,"%Y-%m") as Months
from  orders o
inner join restaurants rt
on o.r_id=rt.r_id
group by rt.r_name,Months
)
select *
from resturant_monthly_sale
where Total_sale>500;
```

Result

Result Grid | Filter Rows: | Exp

r_name	Total_sale	Months
dominos	1000	2022-05
dominos	950	2022-06
dominos	1100	2022-07
kfc	645	2022-05
kfc	990	2022-06
kfc	1935	2022-07
Dosa Plaza	780	2022-05

Finding #5

Show all orders with order details for a particular customer in a particular date range

Query:

```
-- let customer name ='Vartika' and date range is
between "2022-05-10" and '2022-05-22'
with specific_order_detail as
(
select od.*,u.name as Customer_name,o.date as
Order_date
from orders o
inner join order_details od
on o.order_id=od.order_id
join users u
on u.user_id=o.user_id
)
select *
from specific_order_detail
where Customer_name="Vartika"
and (Order_date>"2022-05-10" and
Order_date<'2022-05-22');
```

Result

Result Grid | Filter Rows: | Export: | Wrap

id	order_id	f_id	Customer_name	Order_date
24	1012	8	Vartika	2022-05-20

Finding #6

1. **Find restaurants with max repeated customers**

Query:

```
select rt.r_name,
count(distinct(o.user_id)) as
Repeated_customers
from orders o
join restaurants rt
on rt.r_id=o.r_id
group by rt.r_name
order by Repeated_customers
desc
limit 1;
```

Result

Result Grid | Filter Rows:

r_name	Repeated_customers
dominos	4

Finding #7

1. **Month over month revenue growth of Swiggy**

Query:

```
with total_revenue as
(
select date_format(date,"%Y-%m") as months,
sum(amount) as Total_revenue
from orders
group by months
),
 revenue_growth as(
select * ,
lag(Total_revenue) over (ORDER BY Months) AS
previous_month_revenue
from total_revenue
)
select *,
case
when previous_month_revenue is NULL then null
Else (Total_revenue-
previous_month_revenue/previous_month_revenue*100
)
end as MOM_revenue
from revenue_growth
order by  Months;
```

Result

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

months	Total_revenue	previous_month_revenue	MOM_revenue
2022-05	2425	NULL	NULL
2022-06	3220	2425	3120.0000
2022-07	4845	3220	4745.0000

Finding #8

Customer – favourite food

Query:

```
select
fd.f_name,count(od.order_id)
as total_orders
from food fd
inner join order_details od
on fd.f_id=od.f_id
group by fd.f_name
order by total_orders desc
limit 1;
```

Result

Result Grid | Filter Rows:

f_name	total_orders
Choco Lava cake	13

Finding #9

1. **Find the most loyal customers for all restaurant**

Query:

```
select u.user_id,u.name as
Customer_Name,rt.r_id,rt.r_name
as resturant_name,
count(o.order_id) as Total_orders
from users u
inner join orders o
on o.user_id=u.user_id
join restaurants rt
on rt.r_id=o.r_id
group by
u.user_id,Customer_Name,rt.r_id,r
esturant_name
order by Total_orders desc;
```

Result

Result Grid | Filter Rows: | Export: | Wrap Ce

user_id	Customer_Name	r_id	resturant_name	Total_orders
3	Vartika	2	kfc	3
5	Neha	2	kfc	3
1	Nitish	3	box8	3
4	Ankit	4	Dosa Plaza	3
5	Neha	1	dominos	2
4	Ankit	5	China Town	2
1	Nitish	1	dominos	1

Result 18

Finding #10

Month-over-month revenue growth of a restaurant

Query:

```
with total_revenue as
(
select r_id, date_format(date,"%Y-%m") as months,
sum(amount) as Total_revenue
from orders
group by r_id,months
),
 revenue_growth as(
select * ,
lag(Total_revenue) over (PARTITION BY r_id ORDER BY
Months) AS previous_month_revenue
from total_revenue
)
select *,
case
when previous_month_revenue is NULL then null
Else (Total_revenue-
previous_month_revenue/previous_month_revenue*1
00)
end as MOM_revenue
from revenue_growth
order by r_id,Months;
```

Result

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

r_id	months	Total_revenue	previous_month_revenue	MOM_revenue
1	2022-05	1000	NULL	NULL
1	2022-06	950	1000	850.0000
1	2022-07	1100	950	1000.0000
2	2022-05	645	NULL	NULL
2	2022-06	990	645	890.0000
2	2022-07	1935	990	1835.0000
3	2022-06	480	NULL	NULL

CONCLUSION

The project provides a detailed analysis of Swiggy's operations through a comprehensive examination of its data. Key findings include:

Customer Insights: Identified peak ordering times, customer spending patterns, and preferences for menu items.

Restaurant Performance: Revealed revenue trends and customer satisfaction levels, highlighting areas for improvement and growth.

Platform Performance: Tracked month-over-month revenue growth and assessed delivery efficiency.

RECOMMENDATION

Based on these findings, the project recommends strategies for enhancing customer loyalty, optimizing restaurant operations, and improving overall revenue. These insights aim to support Swiggy in delivering better services, improving customer satisfaction, and driving sustainable growth.