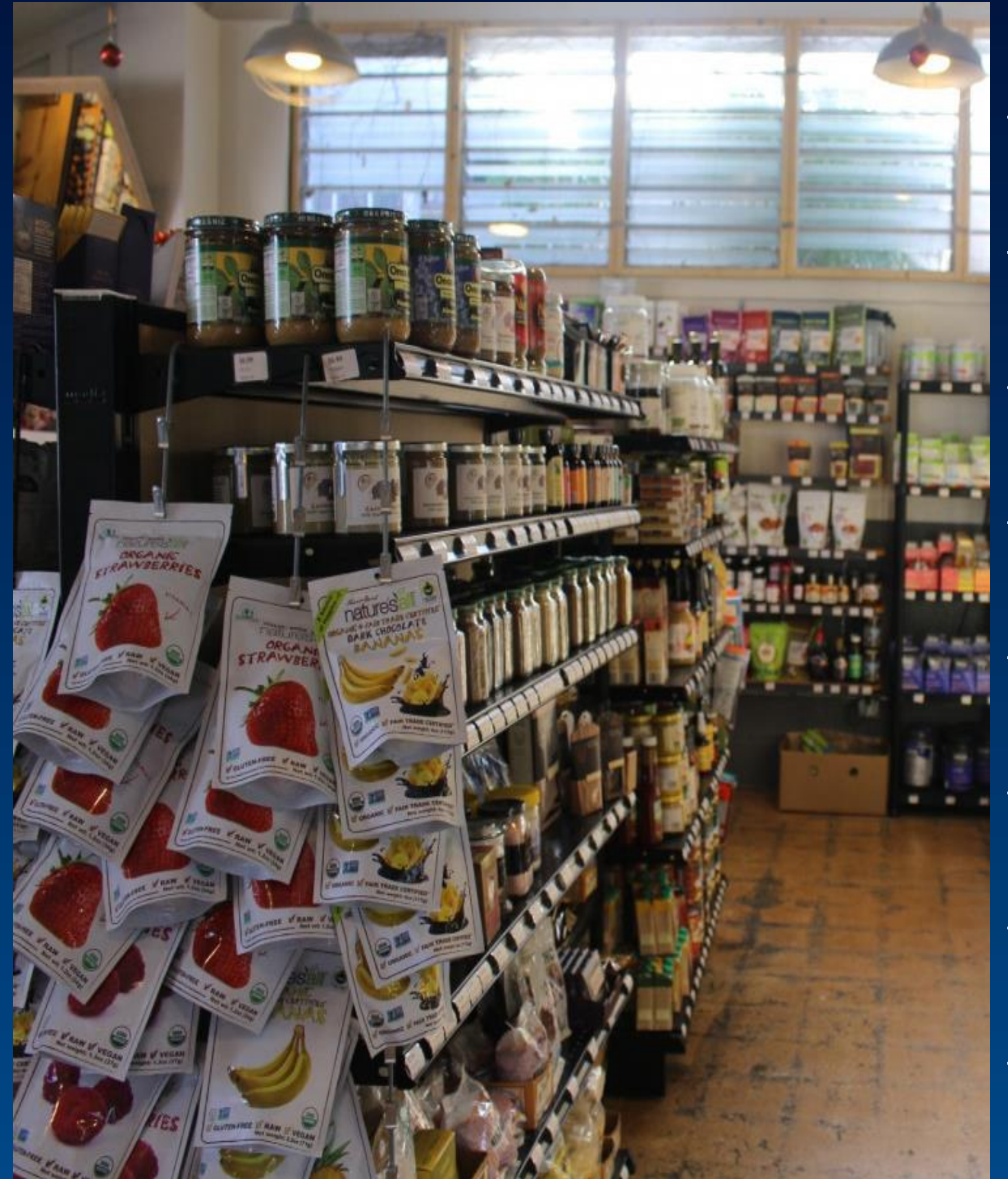


Retail Sales Analyzing

By Pramod Kumar



Project Overview

1.Data Acquisition:

- Data was obtained from the Kaggle API using Jupyter Notebook.
- Initial exploration and preprocessing were performed using pandas.

2.Data Loading to MySQL:

- Processed data was loaded into MySQL from Jupyter Notebook.
- Tables were created in the database with optimized data types for efficient storage.

3.Data Analysis:

- By SQL queries we figure out some questions answer.

Project Workflow:



Top 10 highest revenue product

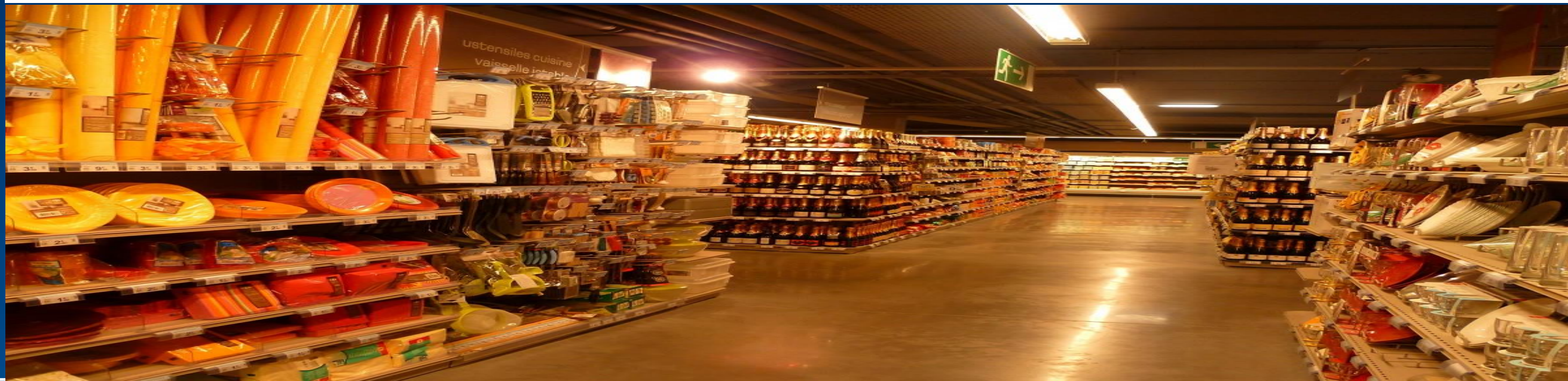
```
SELECT
    product_id, SUM(total_sale) AS sales
FROM
    df_orders
GROUP BY product_id
ORDER BY sales DESC
LIMIT 10;
```

Result Grid			Filter Rows:
	product_id	sales	
▶	TEC-CO-10004722	245056.00	
	OFF-BI-10000545	163777.70	
	TEC-MA-10002412	130406.40	
	FUR-CH-10002024	120090.70	
	TEC-PH-10001459	113041.90	
	TEC-CO-10001449	107388.00	
	OFF-BI-10003527	97082.90	
	TEC-MA-10000822	89622.30	
	FUR-BO-10002213	84014.80	
	TEC-MA-10001047	81549.00	



Top 5 highest selling products in each region

```
with cte as
(select region , product_id, sum(total_sale) as sales from df_orders
  group by region,product_id)
select * from (
select * , dense_rank() over(partition by region order by sales desc) as rn from cte
) as a
where rn <= 5;
```



Result Grid					Filter Rows:	Export:
	region	product_id	sales	rn		
▶	Central	OFF-BI-10000545	125827.50	1		
	Central	TEC-CO-10004722	84875.00	2		
	Central	TEC-MA-10000822	77509.80	3		
	Central	OFF-BI-10001120	55282.50	4		
	Central	OFF-BI-10004995	42210.00	5		
	East	TEC-CO-10004722	106421.00	1		
	East	TEC-MA-10001047	81549.00	2		
	East	FUR-BO-10004834	66364.20	3		
	East	TEC-CO-10001449	60948.00	4		
	East	FUR-CH-10002024	60189.60	5		
	South	TEC-MA-10002412	130406.40	1		
	South	TEC-PH-10001459	73932.10	2		

Find month over month Sales comparision for 2022 and 2023

```
with cte as
(
select year(order_date) as order_year, month(order_date) as order_month, sum(total_sale) as sales from df_orders
group by order_year,order_month
)
select order_month,
sum(case when order_year=2022 then sales else 0 end) as sales_2022,
sum(case when order_year=2023 then sales else 0 end) as sales_2023
from cte
group by order_month
order by order_month;
```

order_month	sales_2022	sales_2023
1	437431.30	434765.50
2	444011.10	731638.80
3	394105.20	393051.90
4	476400.90	543231.50
5	413625.50	410707.90
6	465300.30	328939.00
7	375278.40	422533.70
8	534562.40	465010.30
9	433887.00	420620.50
10	601707.80	626498.30
11	451809.60	334940.60
12	447421.80	491848.90

Find for each category which month had highest sale

```
with cte as
(select category, date_format(order_date,'%Y %M') as order_year_month, sum(total_sale) as sale
from df_orders
group by category, order_year_month)
select category, order_year_month,sale from (
  select * , dense_rank() over(partition by category order by sale desc) as rn from cte ) as g
where rn=1;
```

Result Grid				Filter Rows:	Export:
	category	order_year_month	sale		
▶	Furniture	2023 August	230523.50		
	Office Supplies	2023 February	287244.60		
	Technology	2023 October	295586.50		



Which subcategory had highest growth by profit in 2023 compare to 2022

```
with cte as
(
select sub_category, year(order_date) as order_year, sum(total_profit) as total_profit from df_orders
group by sub_category, order_year
)
,cte2 as(
select sub_category, sum(case when order_year=2022 then total_profit else 0 end) as profit_2022,
sum(case when order_year=2023 then total_profit else 0 end) as profit_2023
from cte
group by sub_category)
select
*, (profit_2023-profit_2022)*100/profit_2022 as growth
from cte2
order by growth desc limit 1;
```

Result Grid |  Filter Rows: | Export:  Wra

	sub_category	profit_2022	profit_2023	growth
▶	Supplies	5161.00	9241.50	79.064135

THANKS!

Do you have any questions?
Ask me in comment the
or
you can also connect with me
on
LINKEDIN

