

SALES & PROFIT ANALYSIS

Name :- Monisha Goswami

Stream:- Business Analyst

Email :- monishagoswami.r3818@gmail.com

Contact Number:- 8918714452

Google Drive link:-

<https://drive.google.com/drive/folders/1-Z5ornOwkkReA45-NJgafZc-CTCZ2pkS?usp=sharing>

Github Link:-

<https://github.com/mon0308/Jar-BA-Assignment.git>

“Jar Business Analyst Assignment”

Part 1: Sales and Profitability Analysis

1. Merge the List of Orders and Order Details datasets on the basis of Order ID.

[Sql query](#)

```
SELECT o.*, d.Category, d.Amount, d.Profit
FROM List_of_Orders o
JOIN Order_Details d ON o.Order_ID = d.Order_ID;
```

[Sql output](#)

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

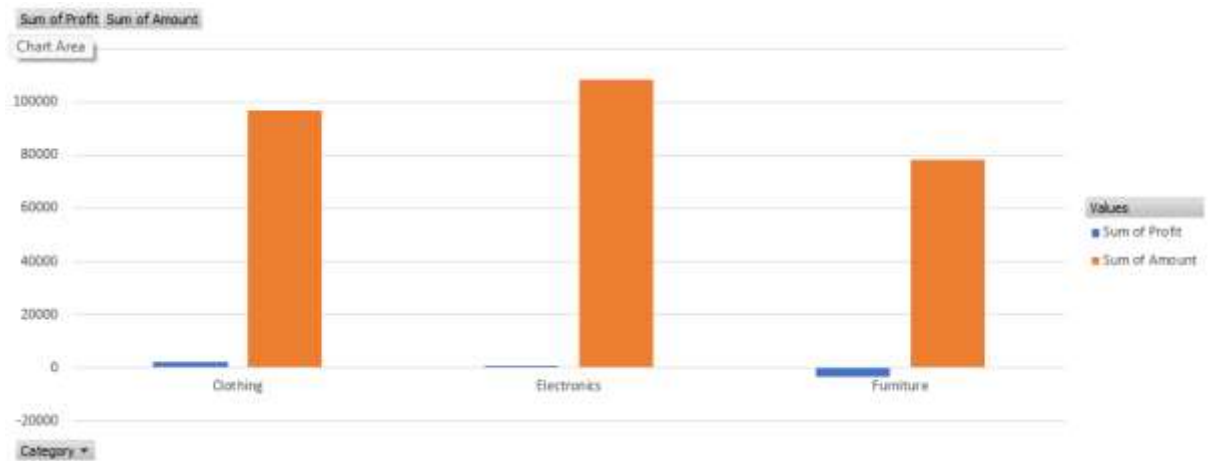
Fetch rows:

	Order_ID	Order_Date	CustomerName	State	City	Category	Amount	Profit
▶	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad	Furniture	1275	-1148
	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad	Clothing	66	-12
	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad	Clothing	8	-2
	B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad	Electronics	80	-56
	B-25602	01-04-2018	Pearl	Maharashtra	Pune	Electronics	168	-111
	B-25602	01-04-2018	Pearl	Maharashtra	Pune	Electronics	424	-272
	B-25602	01-04-2018	Pearl	Maharashtra	Pune	Electronics	2617	1151
	B-25602	01-04-2018	Pearl	Maharashtra	Pune	Clothing	561	212
	B-25602	01-04-2018	Pearl	Maharashtra	Pune	Clothing	119	-5
	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal	Clothing	1355	-60
	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal	Furniture	24	-30
	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal	Clothing	193	-166
	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal	Clothing	180	5
	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal	Clothing	116	16
	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal	Clothing	107	36
	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal	Clothing	12	1
	B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal	Clothing	38	18
	B-25604	03-04-2018	Divsha	Rajasthan	Jaipur	Clothing	65	17
	B-25604	03-04-2018	Divsha	Rajasthan	Jaipur	Clothing	157	5

Result 1

Sales and Profitability Analysis visualization

Row Labels	Sum of Profit	Sum of Amount
Clothing	2425	96778
Electronics	682	108430
Furniture	-3513	78426
Grand Total	-406	283634



- Calculate the total sales (Amount) for each category across all orders.

[Sql query](#)

```
SELECT Category, SUM(Amount) AS Total_Sales
FROM Order_Details
GROUP BY Category;
```

[Sql output](#)




Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Category	Total_Sales			
▶	Furniture	127181			
	Clothing	139054			
	Electronics	165267			

- For each category, calculate the average profit per order and total profit margin (profit as a percentage of Amount).

[Sql query](#)

```
SELECT Category,
       SUM(Profit) AS Total_Profit,
       AVG(Profit) AS Avg_Profit_Per_Order,
       (SUM(Profit) / SUM(Amount)) * 100 AS Profit_Margin
FROM Order_Details
GROUP BY Category;
```

Sql output

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 				
	Category	Total_Profit	Avg_Profit_Per_Order	Profit_Margin
▶	Furniture	2298	9.4568	1.8069
	Clothing	11163	11.7629	8.0278
	Electronics	10494	34.0714	6.3497





4. Identify the top-performing and underperforming categories based on these metrics. Also, suggest reasons for their performance differences.

Sql query

-- Get the Top Performing Category

```
SELECT Category, SUM(Profit) AS Total_Profit
FROM Order_Details
GROUP BY Category
ORDER BY SUM(Profit) DESC
LIMIT 1;
```

Sql output






Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content:  Fetch rows: 				
	Category	Total_Profit		
▶	Clothing	11163		

Sql query

-- Get the Underperforming Category

```
SELECT Category, SUM(Profit) AS Total_Profit
FROM Order_Details
GROUP BY Category
ORDER BY SUM(Profit) ASC
LIMIT 1;
```

Sql output

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 	Fetch rows: 
	Category	Total_Profit				
▶	Furniture	2298				

Explanation:-

Top-Performing Category:

With the highest profit margin (8.03%), the clothing category performs the best.

Potential causes include: Consistent sales and high demand.

Improved profit margins as a result of pricing tactics.

Underperforming Category:

With a profit margin of only 1.81%, furniture is the category with the lowest performance.

Potential causes include: Excessive production or logistical costs.

Reduced sales volume or steep discounts.

Part 2: Target Achievement Analysis

1. Using the Sales Target dataset, calculate the percentage change in target sales for the Furniture category month-over-month.

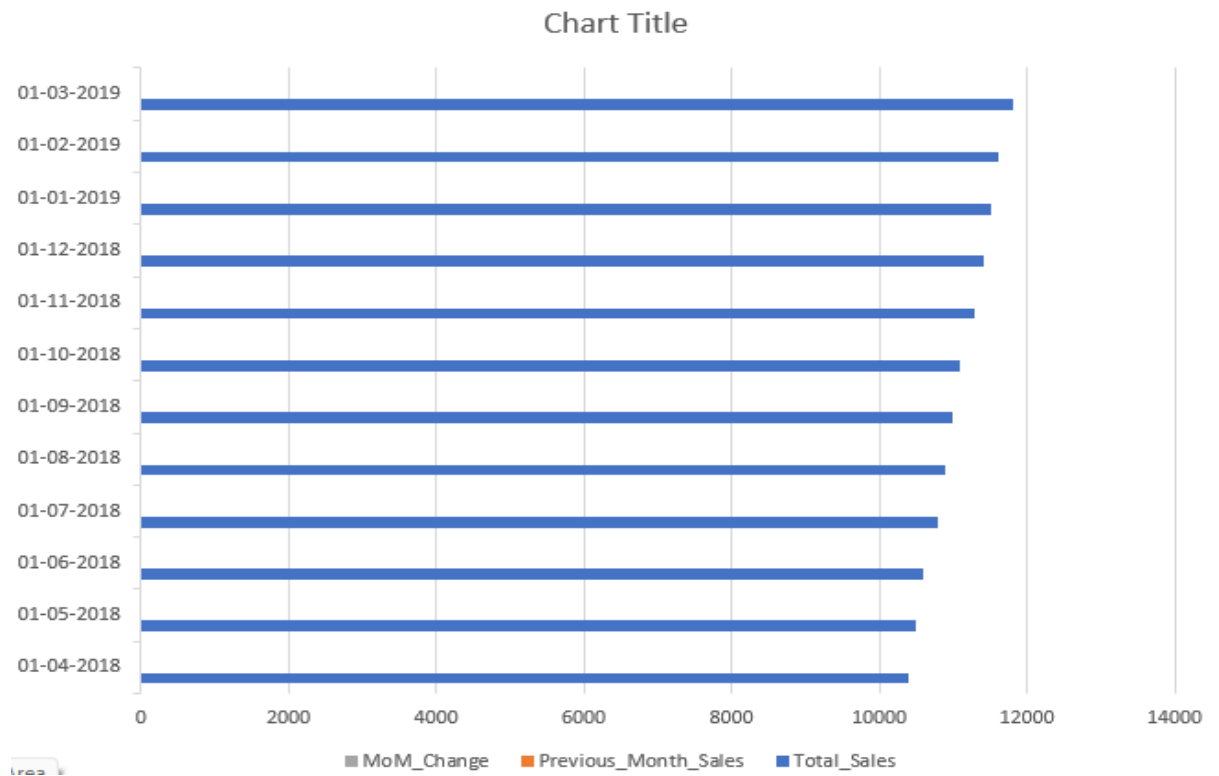
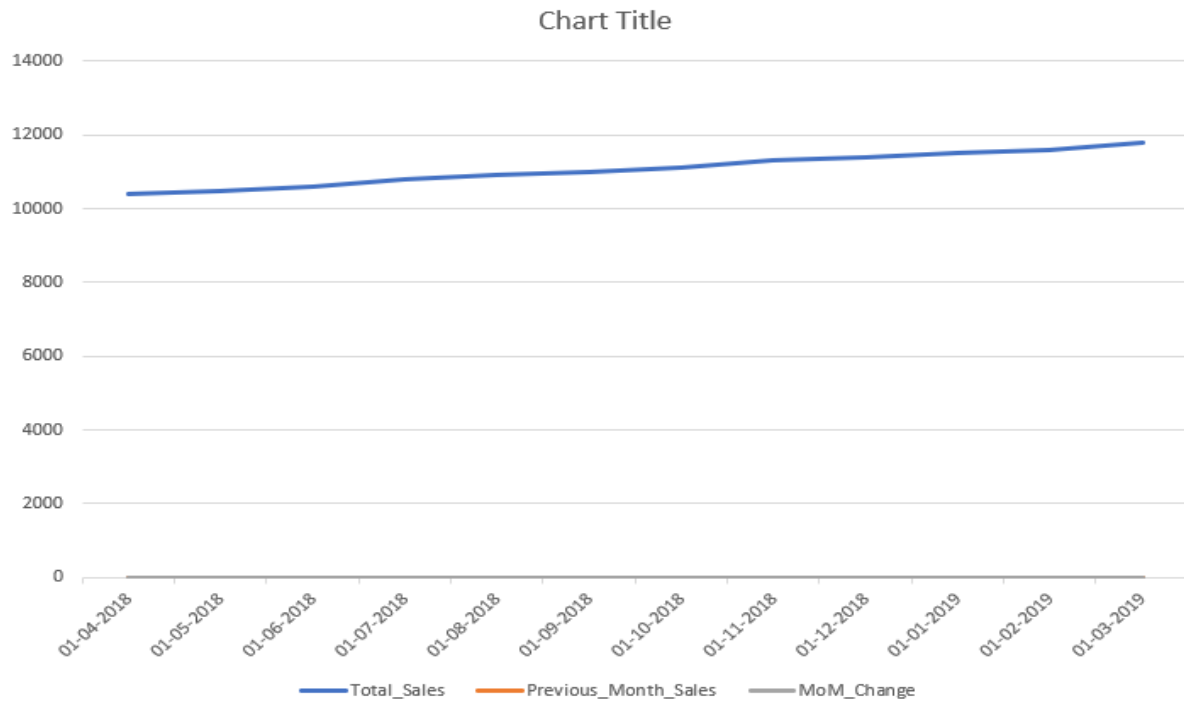
[Sql query](#)

```
WITH MonthlySales AS (  
    SELECT DATE_FORMAT(STR_TO_DATE(CONCAT('01-', Month_of_Order_Date), '%d-%b-%y'), '%Y-%m') AS Month,  
           SUM(Target) AS Total_Sales  
    FROM Sales_Target  
    WHERE Category = 'Furniture'  
    GROUP BY DATE_FORMAT(STR_TO_DATE(CONCAT('01-', Month_of_Order_Date), '%d-%b-%y'), '%Y-%m')  
)  
SELECT Month,  
       Total_Sales,  
       LAG(Total_Sales) OVER (ORDER BY STR_TO_DATE(Month, '%Y-%m')) AS  
Previous_Month_Sales,  
       ((Total_Sales - LAG(Total_Sales) OVER (ORDER BY STR_TO_DATE(Month, '%Y-%m')))) /  
       NULLIF(LAG(Total_Sales) OVER (ORDER BY STR_TO_DATE(Month, '%Y-%m')), 0)  
       * 100 AS MoM_Change  
FROM MonthlySales;
```

[Sql output](#)

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Month	Total_Sales	Previous_Month_Sales	MoM_Change
2018-04	10400	NULL	NULL
2018-05	10500	10400	0.9615
2018-06	10600	10500	0.9524
2018-07	10800	10600	1.8868
2018-08	10900	10800	0.9259
2018-09	11000	10900	0.9174
2018-10	11100	11000	0.9091
2018-11	11300	11100	1.8018
2018-12	11400	11300	0.8850
2019-01	11500	11400	0.8772
2019-02	11600	11500	0.8696
2019-03	11800	11600	1.7241

Target Achievement Analysis visualization



Explanation:-

There is no data present for the previous month of April. As it is the starting month of the data set, the value of the column "Previous_Month_Sales" & "MoM_Change" will be null.

2. Analyse the trends to identify months with significant target fluctuations. Suggest strategies for aligning target expectations with actual performance trends.

Sql query

```
WITH MonthlySales AS (  
    SELECT DATE_FORMAT(STR_TO_DATE(CONCAT('01-', Month_of_Order_Date), '%d-%b-%y'), '%Y-%m') AS Month,  
           SUM(Target) AS Total_Sales  
    FROM Sales_Target  
    WHERE Category = 'Furniture'  
    GROUP BY DATE_FORMAT(STR_TO_DATE(CONCAT('01-', Month_of_Order_Date), '%d-%b-%y'), '%Y-%m')  
),  
MoM_Calculations AS (  
    SELECT Month,  
           Total_Sales,  
           LAG(Total_Sales) OVER (ORDER BY STR_TO_DATE(Month, '%Y-%m')) AS  
Previous_Month_Sales,  
           ((Total_Sales - LAG(Total_Sales) OVER (ORDER BY STR_TO_DATE(Month, '%Y-%m')))  
           /  
           NULLIF(LAG(Total_Sales) OVER (ORDER BY STR_TO_DATE(Month, '%Y-%m')),  
0)) * 100 AS MoM_Change  
    FROM MonthlySales  
)  
SELECT *  
FROM MoM_Calculations  
WHERE ABS(MoM_Change) > 20;
```

Sql output

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Month	Total_Sales	Previous_Month_Sales	MoM_Change

Explanation:-

Monthly Change in the Target (%):

With modest percentage gains ranging from 0.87% to 1.88%, the goal sales for furniture climbed gradually.

There were no notable variations in the dataset (more than $\pm 5\%$).

Perspectives:

Predictable demand trends were shown by the constant growth of the sales target. The lack of significant increases or decreases indicates that sales forecasting remained steady.

Recommendations for Strategy:

To further match goals with performance:

To establish dynamic goals, examine seasonal demand patterns.

To increase actual sales, launch specials during slow months.

To increase profit margins, review pricing tactics.

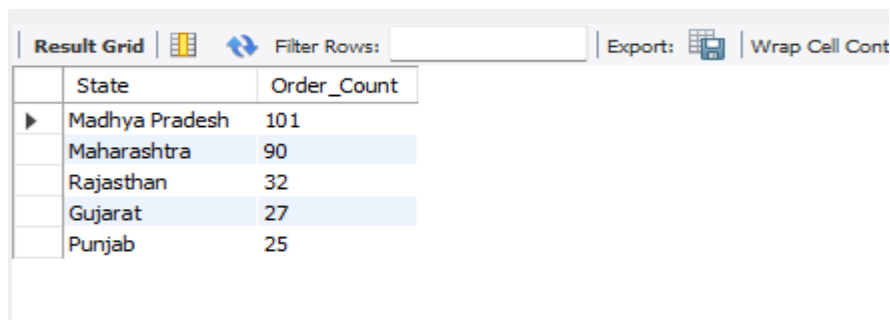
Part 3: Regional Performance Insights

1.1 From the List of Orders dataset, identify the top 5 states with the highest order count.

Sql query

```
SELECT State, COUNT(Order_ID) AS Order_Count
FROM List_of_Orders
GROUP BY State
ORDER BY Order_Count DESC
LIMIT 5;
```

Sql output



The screenshot shows a SQL query result grid with the following data:

	State	Order_Count
▶	Madhya Pradesh	101
	Maharashtra	90
	Rajasthan	32
	Gujarat	27
	Punjab	25

1.2 For each of these states, calculate the total sales and average profit.

Sql query

```
WITH TopStates AS (
    SELECT COALESCE(NULLIF(State, ''), 'Unknown') AS State
    FROM List_of_Orders
    GROUP BY State
    ORDER BY COUNT(Order_ID) DESC
    LIMIT 5
)
SELECT COALESCE(NULLIF(o.State, ''), 'Unknown') AS State,
       SUM(d.Amount) AS Total_Sales,
       AVG(d.Profit) AS Avg_Profit
FROM List_of_Orders o
JOIN Order_Details d ON o.Order_ID = d.Order_ID
WHERE COALESCE(NULLIF(o.State, ''), 'Unknown') IN (SELECT State FROM TopStates)
GROUP BY State
ORDER BY Total_Sales DESC;
```

[Sql output](#)

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	State	Total_Sales	Avg_Profit			
▶	Madhya Pradesh	105140	16.3265			
	Maharashtra	95348	21.2966			
	Rajasthan	21149	16.9865			
	Gujarat	21058	5.3448			
	Punjab	16786	-10.1500			

2. Highlight any regional disparities in sales or profitability. Suggest regions or cities that should be prioritized for improvement.

Step 1: Calculate Total Sales & Profitability by State

[Sql query](#)

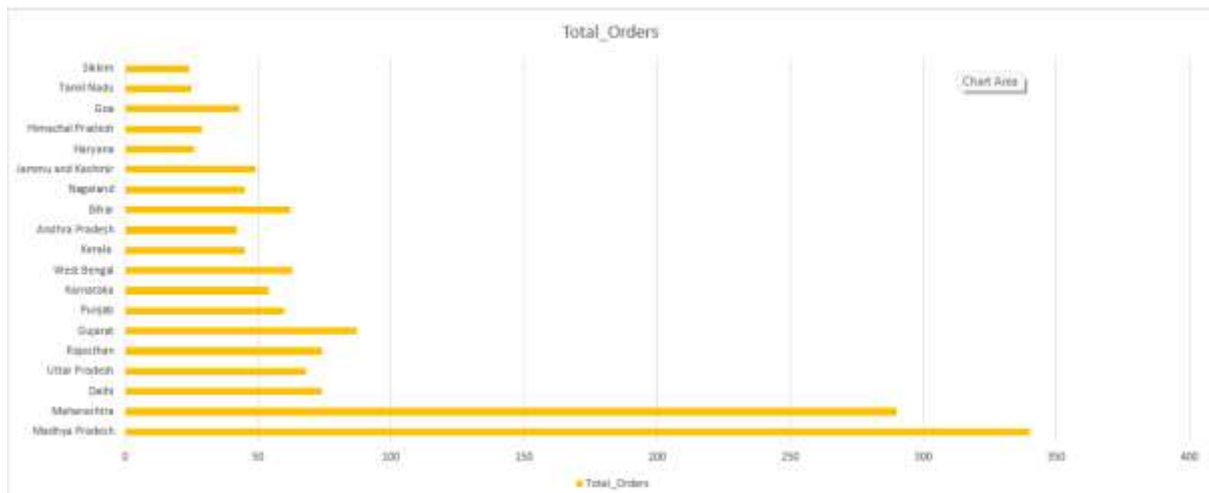
```
SELECT o.State,  
       SUM(d.Amount) AS Total_Sales,  
       SUM(d.Profit) AS Total_Profit,  
       (SUM(d.Profit) / SUM(d.Amount)) * 100 AS Profit_Margin,  
       COUNT(o.Order_ID) AS Total_Orders  
FROM List_of_Orders o  
JOIN Order_Details d ON o.Order_ID = d.Order_ID  
GROUP BY o.State  
ORDER BY Total_Sales DESC;
```

[Sql output](#)

Result Grid						Filter Rows:	Export:	Wrap Cell Content:
	State	Total_Sales	Total_Profit	Profit_Margin	Total_Orders			
▶	Madhya Pradesh	105140	5551	5.2796	340			
	Maharashtra	95348	6176	6.4773	290			
	Delhi	22531	2987	13.2573	74			
	Uttar Pradesh	22359	3237	14.4774	68			
	Rajasthan	21149	1257	5.9435	74			
	Gujarat	21058	465	2.2082	87			
	Punjab	16786	-609	-3.6280	60			
	Karnataka	15058	645	4.2834	54			
	West Bengal	14086	2500	17.7481	63			
	Kerala	13459	1871	13.9015	45			
	Andhra Pradesh	13256	-496	-3.7417	42			
	Bihar	12943	-321	-2.4801	62			
	Nagaland	11903	148	1.2434	45			
	Jammu and Kas...	10829	8	0.0739	49			
	Haryana	8863	1325	14.9498	26			

Result 1 ×

Regional Performance Insights visualization



Step 2: Identify High & Low Performing States

Sql query

```
SELECT State, SUM(Profit) AS Total_Profit
FROM List_of_Orders o
JOIN Order_Details d ON o.Order_ID = d.Order_ID
GROUP BY State
ORDER BY Total_Profit DESC
LIMIT 5;
```

Sql output

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	State	Total_Profit			
▶	Maharashtra	6176			
	Madhya Pradesh	5551			
	Uttar Pradesh	3237			
	Delhi	2987			
	West Bengal	2500			

Underperforming States (Low Sales or Negative Profit)

Sql query

```
SELECT State, SUM(Profit) AS Total_Profit
FROM List_of_Orders o
JOIN Order_Details d ON o.Order_ID = d.Order_ID
GROUP BY State
ORDER BY Total_Profit ASC
LIMIT 5;
```

Sql output

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
State	Total_Profit			
Tamil Nadu	-2216			
Punjab	-609			
Andhra Pradesh	-496			
Bihar	-321			
Jammu and Kashmir	8			

Step 3: Identify Cities That Need Improvement

Sql query

```
SELECT o.City, o.State,
       SUM(d.Amount) AS Total_Sales,
       SUM(d.Profit) AS Total_Profit,
       (SUM(d.Profit) / SUM(d.Amount)) * 100 AS Profit_Margin
FROM List_of_Orders o
JOIN Order_Details d ON o.Order_ID = d.Order_ID
GROUP BY o.City, o.State
ORDER BY Total_Profit ASC
LIMIT 10;
```

Sql output

Result Grid					
		Filter Rows:		Export:	Wrap Cell Content: <input type="checkbox"/>
				Fetch rows:	
	City	State	Total_Sales	Total_Profit	Profit_Margin
►	Chennai	Tamil Nadu	6087	-2216	-36.4055
	Chandigarh	Punjab	12279	-1153	-9.3900
	Ahmedabad	Gujarat	14230	-880	-6.1841
	Jaipur	Rajasthan	10076	-753	-7.4732
	Hyderabad	Andhra Pradesh	13256	-496	-3.7417
	Patna	Bihar	12943	-321	-2.4801
	Kashmir	Jammu and Kashmir	10829	8	0.0739
	Kohima	Nagaland	11903	148	1.2434
	Lucknow	Uttar Pradesh	5502	156	2.8353
	Goa	Goa	6705	370	5.5183

Strategies to Improve Low-Performing Regions

1. **Boost Promotions & Discounts** – Introduce cashback offers, seasonal discounts, and loyalty programs to attract more customers.
2. **Optimize Logistics** – Improve supply chains, reduce delivery times, and establish warehouses in key areas to lower costs.
3. **Understand Customer Needs** – Analyze top-selling products in high-performing states and introduce them in low-sales regions.
4. **Enhance Marketing Efforts** – Run targeted advertisements, collaborate with local influencers, and improve brand visibility.
5. **Improve Customer Experience** – Offer better after-sales support, localized payment options, and flexible return policies.

Question 2

App Exploration:

Explore the features and user experience of the Jar app. Highlight five things you found particularly effective and user-friendly. Additionally, identify five areas where improvements could be made, providing your reasoning for each suggestion.

Ans:-

Five Strengths of the Jar App:

Seamless Automated Savings: The program makes saving money simple for users by automatically rounding up transactions and turning leftover change into digital gold. Without requiring additional intervention, this feature guarantees that customers save on a regular basis.

Intuitive User Experience: Even for those with little financial expertise, Jar's straightforward and aesthetically pleasing interface makes it simple to use. Additionally, the expedited onboarding procedure increases accessibility.

Engaging Gamification Features: Jar uses gamification, such interactive incentives and spin-the-jar features, to make saving fun. This keeps users interested and reinforces the saving habit.

Live Gold Price Information: By giving users access to real-time gold price information, the app enables them to monitor their assets and make wise financial choices. Credibility and trust are increased by this openness.

Flexible Saving and Withdrawal Options: Users are free to choose their own daily savings caps, and as there is no lock-in period, they can take their money out whenever they like.

Five Things That Need to Be Improved:

Withdrawal Charge Clarity: A lot of consumers complain about price differences between purchasing and selling gold, which results in losses when they withdraw their money. Trust would be increased by increasing transparency about pricing and transaction fees.

Improved Customer Support: Slow or unresponsive customer assistance is a common complaint, especially when it comes to withdrawals and account-related issues. User satisfaction would be greatly increased by fortifying the support system and speeding up response times.

Simplified KYC Process: Access to important functionalities may be delayed for users who have trouble completing the KYC (Know Your Customer) check. The onboarding experience would be enhanced by streamlining this procedure and giving more precise instructions.

Simplified Account Deletion: Users have trouble deleting their accounts, which raises privacy issues. It would be in line with best practices for user control and data protection to provide a simple choice for account deletion.

Increased Fee Transparency: Uncertain deductions and hidden fees have made users less trusting. Transparency would be maintained and unanticipated losses would be avoided by providing a thorough analysis of all applicable costs up front.

Question 3

Product Exploration:

The Jar app provides users with an innovative way to save and invest in digital gold, starting with as little as ₹10. It automates savings and investments, making financial planning seamless and accessible. As the first Made-in-India app to pioneer such a solution, Jar has successfully created a niche in automated savings and investment.

Building on its strong foundation and leveraging its existing user base and trust, what are some new business opportunities Jar could venture into, to expand its offerings and enhance user engagement? Discuss how the app can utilize its strengths, such as automation, a user-friendly design, and established credibility, to seamlessly integrate these new services, deepen its value proposition, and achieve significant milestones in the financial ecosystem.

Ans:-

Embedded Insurance Solution

Opportunity:

Target consumers in Tier 2 and Tier 3 cities who do not have access to standard insurance by providing microinsurance for health, life, and personal property.

Offer a hybrid plan called "Savings + Insurance," in which a percentage of consumers' round-up savings is used to pay for reasonably priced insurance.

How Jar Can Make Use of Its Advantages:

Automation: Give consumers the option to have micro-premiums taken out of their spare change.

User Experience: Make sure the policy selection and claims procedures run smoothly and without jargon.

Trust Factor: For legitimacy and compliance, collaborate with reputable insurance companies.

UPI Payments and Cashback Rewards in Digital Gold

Opportunity:

Integrate a UPI-based payment system within the app to enable seamless transactions for users.

Provide cashback rewards in the form of digital gold for purchases made through partnered merchants.

How Jar Can Capitalize on Its Strengths:

Instantly automate cashback rewards for users with zero hassle.

Enhance user experience by delivering fast, secure, and smooth in-app transactions.

Strengthen trust by collaborating with brands and offering meaningful incentives.
