Chinmay Raut

🔗 LinkedIn 💻 GitHub 📄 Resume

📌 **Assignment Overview**
In this assignment, I utilized various tools and technologies to conduct a comprehensive analysis, ensuring accuracy and clarity in insights.

🛠 **Tools & Technologies Used:**

🔹 **Python & Data Libraries**: Leveraged Pandas for data manipulation and Matplotlib/Seaborn for visualization to make the data more interpretable.
🔹 **Draw.io**: Designed detailed workflows and process diagrams to visually map out data processes and analytical structures.
🔹 **Large Language Models (LLMs)**: Assisted in refining the report structure, improving clarity, and enhancing coherence.

Additionally, I conducted thorough research on the **Jar app**, analyzing its features, user experience, and market positioning. This holistic approach allowed me to develop well-informed insights and actionable recommendations.

✅ **Initial Data Quality Check**

Before diving into the analysis, I conducted a preliminary data quality assessment using Python's built-in functions to ensure a reliable dataset:

🔍 **Data Cleaning**: Identified and handled missing values or inconsistencies in the dataset.
🏷 **Labeling & Data Types**: Verified column labels and ensured data types were correctly formatted for accurate analysis.

This foundational step was crucial in maintaining data integrity and ensuring precise, actionable insights in the subsequent analysis.

📊 **Understanding the Data**
The dataset consists of three files, each providing crucial insights into order details, sales, and targets.

📁 **Dataset 1: List_of_Orders_55FFC79CF8.csv**

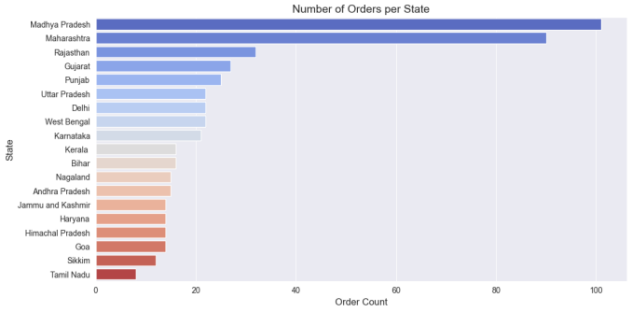
📌 **Overview:** Contains order-related data with the following structure:

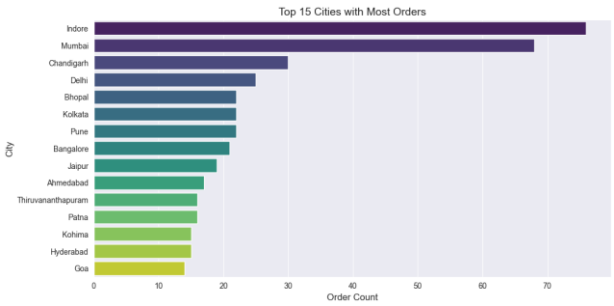
📄 **Columns:**
✓ Order ID
✓ Order Date
✓ Customer Name
✓ State
✓ City

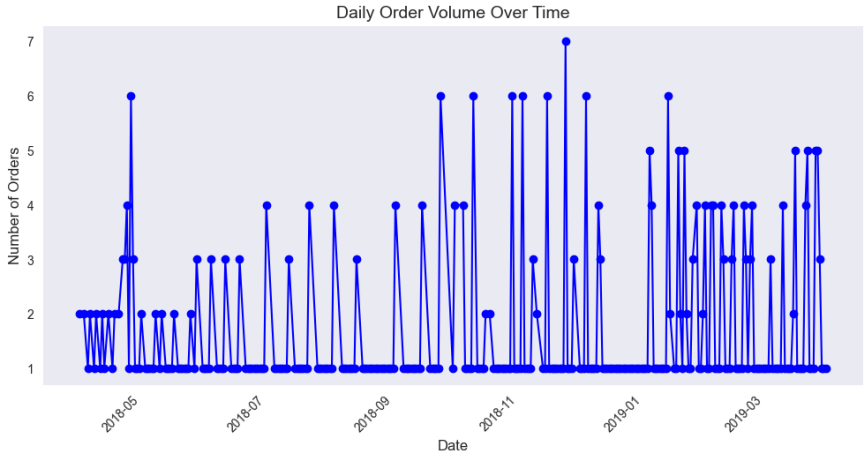
✅ **Data Consistency:**
The dataset is well-structured, with unique Order IDs ensuring data integrity.

List_of_Orders_55FFC79CF8.csv

Order ID	Order Date	Customer Name	State	City
B-25601	01-04-2018	Bharat	Gujarat	Ahmedabad
B-25602	01-04-2018	Pearl	Maharashtra	Pune
B-25603	03-04-2018	Jahan	Madhya Pradesh	Bhopal
B-25604	03-04-2018	Divsha	Rajasthan	Jaipur
B-25605	05-04-2018	Kasheen	West Bengal	Kolkata
B-25606	06-04-2018	Hazel	Karnataka	Bangalore
B-25607	06-04-2018	Sonakshi	Jammu and Kashmir	Kashmir







📊 Exploratory Data Analysis (EDA)

To understand the dataset better, I created several visualizations:

📈 **Number of Orders per State** – Shows order distribution across different states.
🏙️ **Top Cities by Order Volume** – Highlights cities with the highest order counts.
🗓️ **Daily Order Volume Over Time** – Identifies trends and patterns over time.

These insights provide a high-level overview of the data, helping to spot key trends before deeper analysis.

📁 **Dataset 2: Order_Details_19795F61CF.csv**

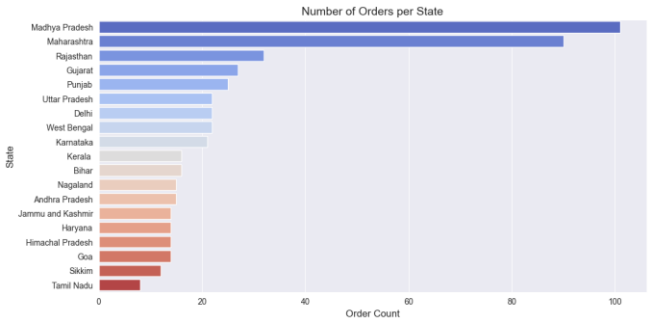
📌 **Overview:** Contains detailed order information with the following columns:

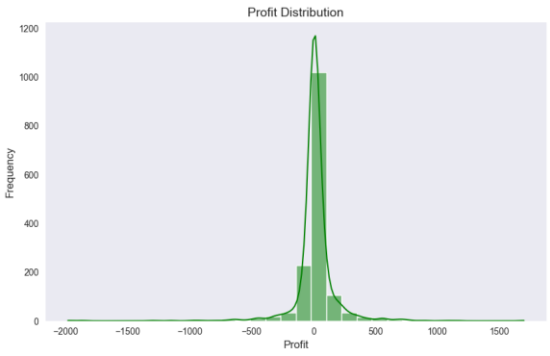
✓ **Order ID** – Unique identifier for each order
💰 **Amount** – Total sales amount for the order
📈 **Profit** – Profit or loss associated with the order
📦 **Quantity** – Number of items purchased
🏷 **Category** – Broad classification of products
📌 **Sub-Category** – Specific product type within a category

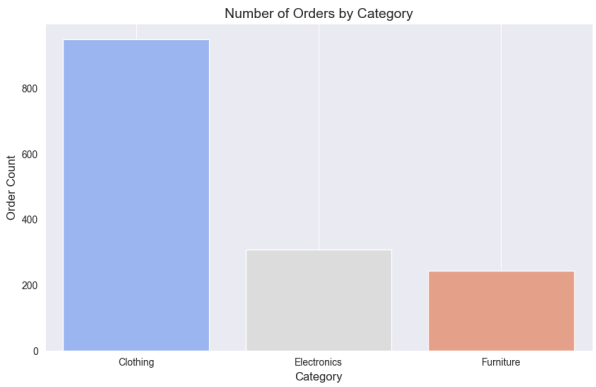
📊 **Exploratory Data Analysis (EDA) with Visualizations**

Order_Details_19795F61CF.csv

Order ID	Amount	Profit	Quantity	Category	Sub-Category
B-25601	1275	-1148	7	Furniture	Bookcases
B-25601	66	-12	5	Clothing	Stole
B-25601	8	-2	3	Clothing	Handkerchief
B-25601	80	-56	4	Electronics	Electronic Games







To extract meaningful insights, I created these graphs:

📍 **Number of Orders per State** – Shows distribution of orders across states.
📊 **Profit Distribution** – Visualizes overall profit/loss trends.
📦 **Orders per Category** – Highlights product category-wise order volume.
📉 **Sales Amount vs. Profit (Scatter Chart)** – Analyzes sales and profit relationship, identifying trends and potential outliers.

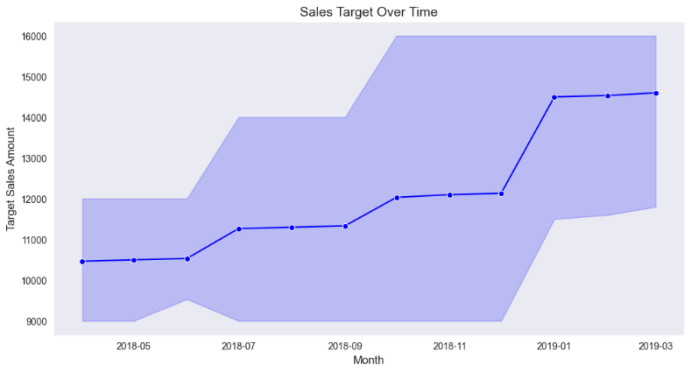
📁 **Dataset 3: Sales_target_DD2E9B96A0.csv**

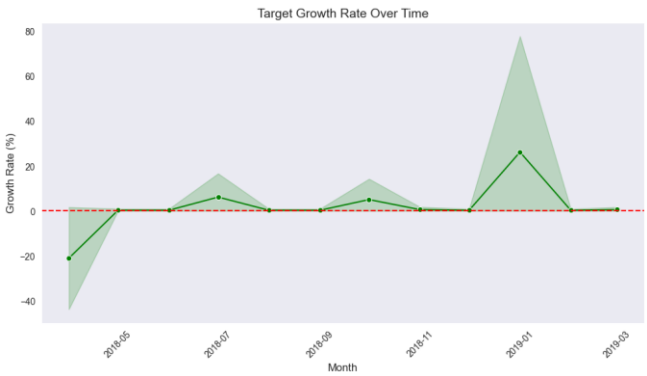
📌 **Overview:** Contains monthly sales targets for different product categories.

🗓️ **Month of Order Date** – Month and year of the sales target.
🏷️ **Category** – Product category for which the sales target is set.
🎯 **Target** – Sales target amount for the given month and category.

This dataset helps in evaluating how actual sales compare to projected targets, providing key business insights.

Month of Order Date	Category	Target
Apr-18	Furniture	10,400.00
May-18	Furniture	10,500.00
Jun-18	Furniture	10,600.00
Jul-18	Furniture	10,800.00
Aug-18	Furniture	10,900.00
Sep-18	Furniture	11,000.00
Oct-18	Furniture	11,100.00





Question 1 : (30 marks)

Sales Analysis:

Part 1: Sales and Profitability Analysis

1.1 Merge the List of Orders and Order Details datasets on the basis of Order ID. Calculate the total sales (Amount) for each category across all orders.

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

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

🔁 1.1 Merging Datasets & Calculating Total Sales by Category

To perform a comprehensive analysis, we merged the **List of Orders** and **Order Details** datasets on **Order ID**, consolidating order details, sales amount, and profitability.

```
# Ensure consistent date format for List of Orders ('Order Date')
orders['Order Date'] = pd.to_datetime(orders['Order Date'], format="%d-%m-%Y")
orders['Order Date'] = orders['Order Date'].dt.strftime("%d-%m-%Y")
```

```
Orders DF
  Order ID  Order Date CustomerName          State       City
0  B-25601  01-04-2018       Bharat        Gujarat  Ahmedabad
1  B-25602  01-04-2018        Pearl    Maharashtra       Pune
2  B-25603  03-04-2018        Jahan  Madhya Pradesh     Bhopal
3  B-25604  03-04-2018       Divsha      Rajasthan     Jaipur
4  B-25605  05-04-2018      Kasheen    West Bengal    Kolkata
====================================
Orders Details DF
  Order ID  Amount  Profit  Quantity     Category      Sub-Category
0  B-25601  1275.0 -1148.0         7    Furniture         Bookcases
1  B-25601    66.0   -12.0         5     Clothing             Stole
2  B-25601     8.0    -2.0         3     Clothing       Hankerchief
3  B-25601    80.0   -56.0         4  Electronics  Electronic Games
4  B-25602   168.0  -111.0         2  Electronics            Phones
====================================
Merged DF
  Order ID  Order Date CustomerName        State       City  Amount  Profit
0  B-25601  01-04-2018       Bharat      Gujarat  Ahmedabad  1275.0 -1148.0
1  B-25601  01-04-2018       Bharat      Gujarat  Ahmedabad    66.0   -12.0
2  B-25601  01-04-2018       Bharat      Gujarat  Ahmedabad     8.0    -2.0
3  B-25601  01-04-2018       Bharat      Gujarat  Ahmedabad    80.0   -56.0
4  B-25602  01-04-2018        Pearl  Maharashtra       Pune   168.0  -111.0
```

💰 1.2 Profit Analysis by Category

For each category, we calculated key profitability metrics:

📊 **Key Metrics:**
✓ **Total Sales (₹)** – Sum of revenue for each category.
✓ **Total Profit (₹)** – Sum of profit for each category.
✓ **Order Count** – Number of unique orders.

✓ **Average Profit per Order (₹)** – Profit divided by total unique orders.
✓ **Profit Margin (%)** – Profit as a percentage of total sales.

Python Code Snippet:

```
avg_profit_results = {}
for cat in cat:
    # Filter the merged dataset for the current category
    subset = merged_df[merged_df['Category'] == cat]

    # Calculate total profit for the category
    total_profit = subset['Profit'].sum()

    # Calculate the number of unique orders in the category
    unique_orders = subset['Order ID'].nunique()

    # Compute average profit per order (handling division by zero)
    avg_profit = total_profit / unique_orders if unique_orders > 0 else 0

    # Store the result
    avg_profit_results[cat] = avg_profit
for cat, avg in avg_profit_results.items():
    print(f"{cat}: {avg}")
```

```
Furniture: 12.35483870967742
Clothing: 28.404580152671755
Electronics: 51.44117647058823
```

```
# what are the categoories in sales_target
cat = sales_target['Category'].unique()
print(cat)
```

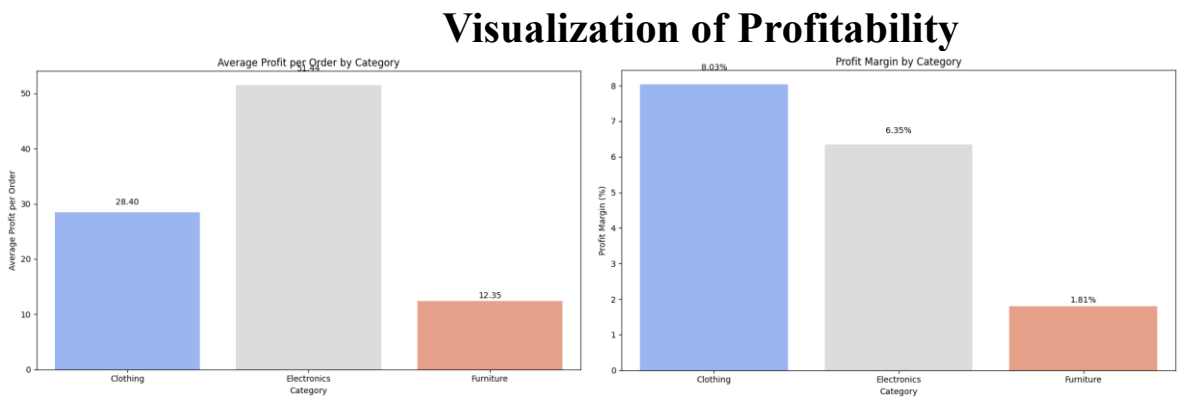
```
['Furniture' 'Clothing' 'Electronics']
```

```
# Group by Category and calculate total sales, total profit, and count of unique orders
category_stats = merged_df.groupby('Category').agg(
    total_sales=('Amount', 'sum'),
    total_profit=('Profit', 'sum'),
    order_count=('Order ID', 'nunique')
).reset_index()
```

```
# sales_target
def calculate_avg_profit(group):
    total_profit = group['Profit'].sum()
    unique_orders = group['Order ID'].nunique()
    # Avoid division by zero just in case
    avg_profit = total_profit / unique_orders if unique_orders > 0 else 0
    return avg_profit
```

📉 Visualization of Profitability

To better understand profitability, I generated graphs to visualize trends in sales and profit margins.



📌 3. Insights & Reasons for Performance Differences

🔝 Top-Performing Categories

☐Based on Average Profit per Order: 🏆 Electronics (₹51.44 per order)

- Higher-priced products contribute more profit per transaction.
- Customers are willing to spend more on premium electronic goods.
- Higher margins on accessories and premium gadgets.

2️⃣Based on Profit Margin (%): ⏳ Clothing (8.03%)

- Lower cost of goods sold (COGS) compared to other categories.
- High volume of sales compensates for the lower ticket size.

- Seasonal trends and branding allow premium pricing.

⚠️ Underperforming Categories

☐Based on Average Profit per Order: ❌ Furniture (₹12.35 per order)

- High storage and shipping costs reduce net profit.
- Customers often wait for discounts before purchasing.
- Longer product lifecycle leads to lower purchase frequency.

2️⃣Based on Profit Margin (%): ❌ Furniture (1.81%)

- High procurement and manufacturing costs.
- Bulkier items lead to expensive shipping and storage.
- Heavy competition with price-sensitive customers.

📌 Recommendations for Improvement

Category	Profit Type	Issues	Possible Solutions
Furniture	Low Profit, Low Margin	High shipping & storage costs	Introduce modular designs, optimize pricing.
Clothing	Moderate Profit, High Margin	Seasonal demand fluctuations	Implement dynamic pricing, leverage sales.
Electronics	High Profit, Moderate Margin	Supply chain dependencies	Focus on premium products, explore subscriptions.

This structured approach ensures a deeper understanding of category-wise performance and helps formulate strategic business decisions.

Part 2: Target Achievement Analysis

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

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

Part 2: Target Achievement Analysis

1. Percentage Change in Target Sales (Month-over-Month)

To understand how sales targets evolved over time, we calculated the month-over-month percentage change for the Furniture category. This metric highlights periods of significant growth or fluctuations in sales targets.

Month of Order Date	Target Sales (₹)	Percentage Change (%)
Apr-18	10,400	—
May-18	10,500	0.96%
Jun-18	10,600	0.95%
Jul-18	10,800	1.89%
Aug-18	10,900	0.93%
Sep-18	11,000	0.92%
Oct-18	11,100	0.91%
Nov-18	11,300	1.80%
Dec-18	11,400	0.88%
Jan-19	11,500	0.87%
Feb-19	11,600	0.87%
Mar-19	11,800	1.72%

---

2. Insights from the Trends

📈 **Overall Growth Pattern**

- The sales targets show a steady upward trend, with a consistent month-over-month increase of around 0.87% to 0.96% in most months.
- This suggests a structured and predictable target-setting approach.

📊 **Notable Spikes in Target Growth**

- **July 2018 (+1.89%)**, **November 2018 (+1.80%)**, and **March 2019 (+1.72%)** stand out as months with a higher-than-usual increase in targets.
- These could be linked to:
    - Seasonal sales trends (e.g., mid-year and year-end demand surges).
    - Festive promotions or holiday sales expectations.
    - Annual or quarterly business strategy shifts, adjusting targets to align with past performance.

📉 **Slower Growth in Certain Months**

- Some months, like **December 2018 to February 2019**, had lower growth rates (~0.87%), possibly due to:
    - Post-festive season slowdowns.
    - Market saturation or economic factors affecting demand.

---

3. Strategies to Align Target Expectations with Performance Trends

✅ **1. Data-Driven Target Adjustments**

- Use historical sales data and market trends to set more realistic and achievable targets.

- Instead of sudden jumps in target growth, aim for a smoother and more consistent increase.

📦 2. Inventory & Logistics Planning

- Anticipate peak sales periods (**July, November, March**) and optimize inventory accordingly.
- Align supply chain and warehouse operations to meet the increased demand without overstocking.

💰 3. Seasonal Promotions & Pricing Strategies

- Launch targeted discounts, promotions, and bundled offers to capitalize on peak months.
- Implement dynamic pricing strategies to encourage sales during low-growth months.

📢 4. Marketing & Customer Engagement

- Increase advertising spend and marketing efforts ahead of high-growth months to maximize sales.
- Focus on customer retention strategies to ensure steady revenue flow even during slow months.

---

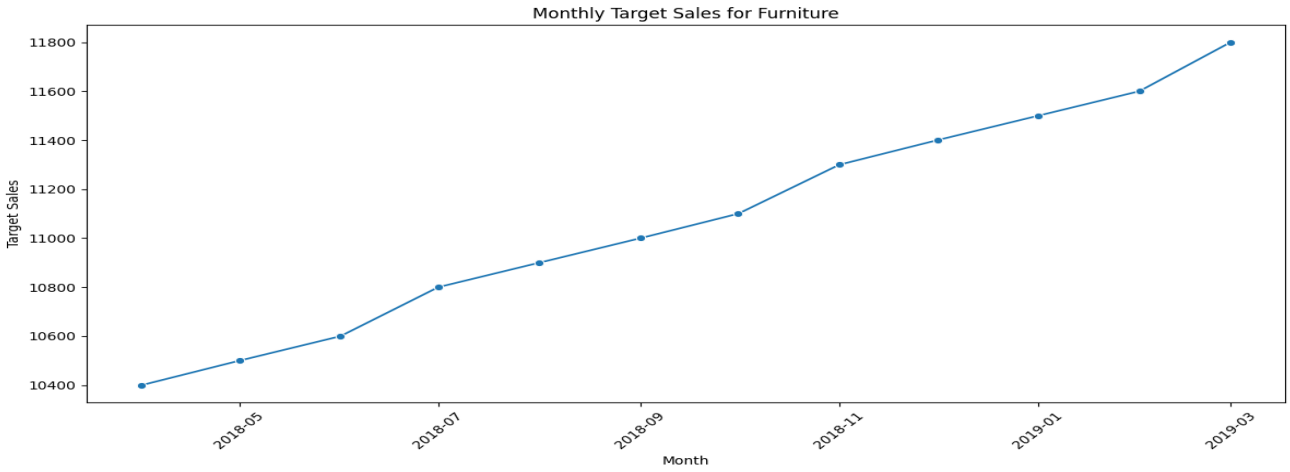
4. Final Takeaway 💡

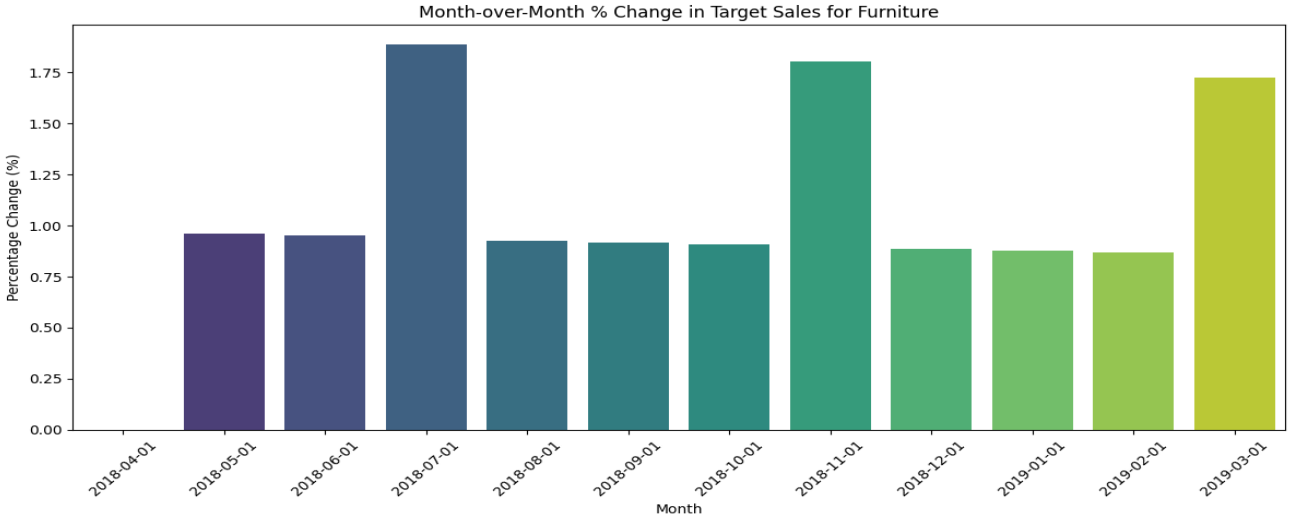
While the steady growth trend in targets indicates a well-planned business strategy, the spikes in certain months suggest opportunities for better forecasting and alignment with actual demand. By implementing data-backed decision-making, inventory optimization, and targeted marketing strategies, the business can enhance profitability and reduce inefficiencies..

Code

```
# Filter for the Furniture category
furniture_targets = sales_target[sales_target['Category'] == 'Furniture'].copy()

# Convert 'Month of Order Date' to datetime (assuming it's in "dd-mm-yyyy" format)
furniture_targets['Month of Order Date'] = pd.to_datetime(furniture_targets['Month of Order Date'], format="%d-%m-%Y")

# Sort the DataFrame by date
furniture_targets.sort_values('Month of Order Date', inplace=True)

# Calculate the month-over-month percentage change in target sales
furniture_targets['Pct_Change'] = furniture_targets['Target'].pct_change() * 100

# Display the updated DataFrame
print(furniture_targets[['Month of Order Date', 'Target', 'Pct_Change']])

# Plot 1: Monthly Target Sales for Furniture
plt.figure(figsize=(12,6))
sns.lineplot(data=furniture_targets, x='Month of Order Date', y='Target', marker='o')
plt.title('Monthly Target Sales for Furniture')
plt.xlabel('Month')
plt.ylabel('Target Sales')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Plot 2: Month-over-Month Percentage Change in Target Sales
plt.figure(figsize=(12,6))
sns.barplot(data=furniture_targets, x='Month of Order Date', y='Pct_Change', palette='viridis')
plt.title('Month-over-Month % Change in Target Sales for Furniture')
plt.xlabel('Month')
plt.ylabel('Percentage Change (%)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Month of Order Date	Target Sales (₹)	Percentage Change (%)
Apr-18	10,400	—
May-18	10,500	0.96%
Jun-18	10,600	0.95%
Jul-18	10,800	1.89%
Aug-18	10,900	0.93%
Sep-18	11,000	0.92%
Oct-18	11,100	0.91%
Nov-18	11,300	1.80%
Dec-18	11,400	0.88%
Jan-19	11,500	0.87%
Feb-19	11,600	0.87%
Mar-19	11,800	1.72%





Part 3: Regional Performance Insights

• From the List of Orders dataset, identify the top 5 states with the highest order count. For each of these states, calculate the total sales and average profit.

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

---

1. Top 5 States with the Highest Order Count

The states with the most orders are:

1**Madhya Pradesh** (101 orders)
2**Maharashtra** (90 orders)
3 **Rajasthan** (32 orders)
4 **Gujarat** (27 orders)
5 **Punjab** (25 orders)

Regional Metrics by State:

	State	order_count	total_sales	total_profit
0	Andhra Pradesh	15	13256.0	-496.0
1	Bihar	16	12943.0	-321.0
2	Delhi	22	22531.0	2987.0
3	Goa	14	6705.0	370.0
4	Gujarat	27	21058.0	465.0
5	Haryana	14	8863.0	1325.0
6	Himachal Pradesh	14	8666.0	656.0
7	Jammu and Kashmir	14	10829.0	8.0
8	Karnataka	21	15058.0	645.0
9	Kerala	16	13459.0	1871.0
10	Madhya Pradesh	101	105140.0	5551.0
11	Maharashtra	90	95348.0	6176.0
12	Nagaland	15	11903.0	148.0
13	Punjab	25	16786.0	-609.0
14	Rajasthan	32	21149.0	1257.0
15	Sikkim	12	5276.0	401.0
16	Tamil Nadu	8	6087.0	-2216.0
17	Uttar Pradesh	22	22359.0	3237.0
18	West Bengal	22	14086.0	2500.0

	avg_profit_per_order	profit_margin
0	-33.066667	-3.741702
1	-20.062500	-2.480105
2	135.772727	13.257290
3	26.428571	5.518270
4	17.222222	2.208187
5	94.642857	14.949791
6	46.857143	7.569813
7	0.571429	0.073876
8	30.714286	4.283437
9	116.937500	13.901479
10	54.960396	5.279627
11	68.622222	6.477325
12	9.866667	1.243384
13	-24.360000	-3.628023
14	39.281250	5.943543
15	33.416667	7.600455
16	-277.000000	-36.405454
17	147.136364	14.477392
18	113.636364	17.748119

These states contribute significantly to the overall sales and profitability, but their performance varies across key metrics.

2. Sales and Profitability Analysis

State	Total Sales (₹)	Avg Profit per Order (₹)	Profit Type
Madhya Pradesh	105,140	16.33	Moderate Profit, Moderate Margin
Maharashtra	95,348	21.30	High Profit, High Margin
Rajasthan	21,149	16.99	Moderate Profit, Moderate Margin
Gujarat	21,058	5.34	Low Profit, Low Margin
Punjab	16,786	-10.15	Loss-Making

- **Maharashtra** leads in profitability, with the highest average profit per order (₹21.30).
- **Punjab** is the weakest performer, showing negative profits despite being in the top 5 for order count.
- **Gujarat** has a low profit margin, indicating inefficiencies in pricing or operational costs.

---

3. Regional Disparities and Key Observations

📌 **Maharashtra & Madhya Pradesh:** ✅ Strong performance in total sales and profitability.
✅ Maharashtra has a high profit margin, suggesting efficient sales strategies.
✅ Madhya Pradesh shows moderate profitability, which can be further optimized with improved logistics.

📌 **Rajasthan & Gujarat:** ⚠️ Lower profitability despite high order volumes.
⚠️ Gujarat has a low average profit per order (₹5.34), indicating possible high operational costs or price undercutting.
⚠️ Rajasthan has a decent profit margin, but its total sales are lower than expected given its order count.

📌 **Punjab:** ❌ Negative profitability (-₹10.15 avg profit per order) suggests major issues.
❌ Possible reasons: high return rates, pricing issues, or inefficiencies in fulfillment costs.

---

4. Recommendations for Regional Improvement

✅ **1. Optimize Pricing & Cost Efficiency (Punjab, Gujarat)**

- Reassess pricing models and consider increasing profit margins on high-demand products.
- Negotiate with suppliers to reduce procurement costs, particularly in Gujarat.

✅ **2. Improve Logistics & Supply Chain (Madhya Pradesh, Rajasthan)**

- Invest in faster and cost-effective logistics to lower delivery costs.
- Identify high-demand cities and establish localized fulfillment centers to reduce overhead costs.

✅ **3. Targeted Marketing & Promotions (Punjab, Gujarat)**

- Offer region-specific discounts and bundle deals to increase sales.
- Conduct customer surveys to identify reasons behind Punjab's negative profitability.

✅ **4. Expand High-Margin Categories (Maharashtra, Madhya Pradesh)**

- Focus on selling premium or high-margin products in well-performing states.
- Consider subscription-based sales models to increase customer retention.

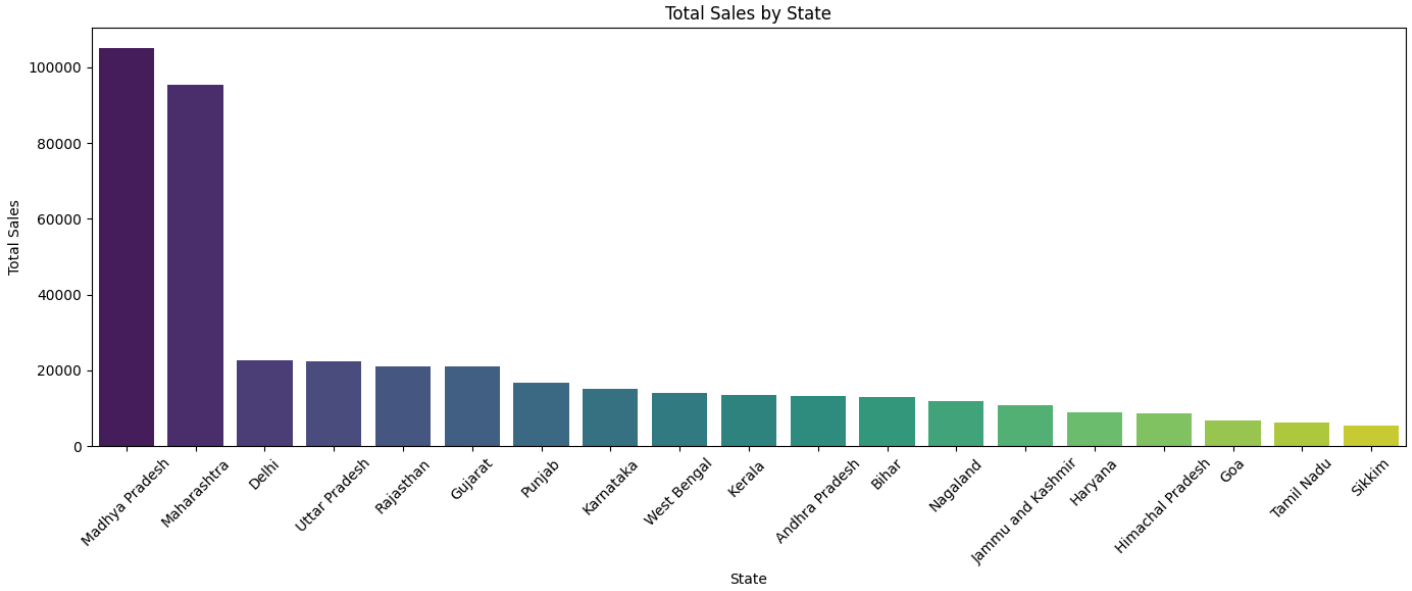
---

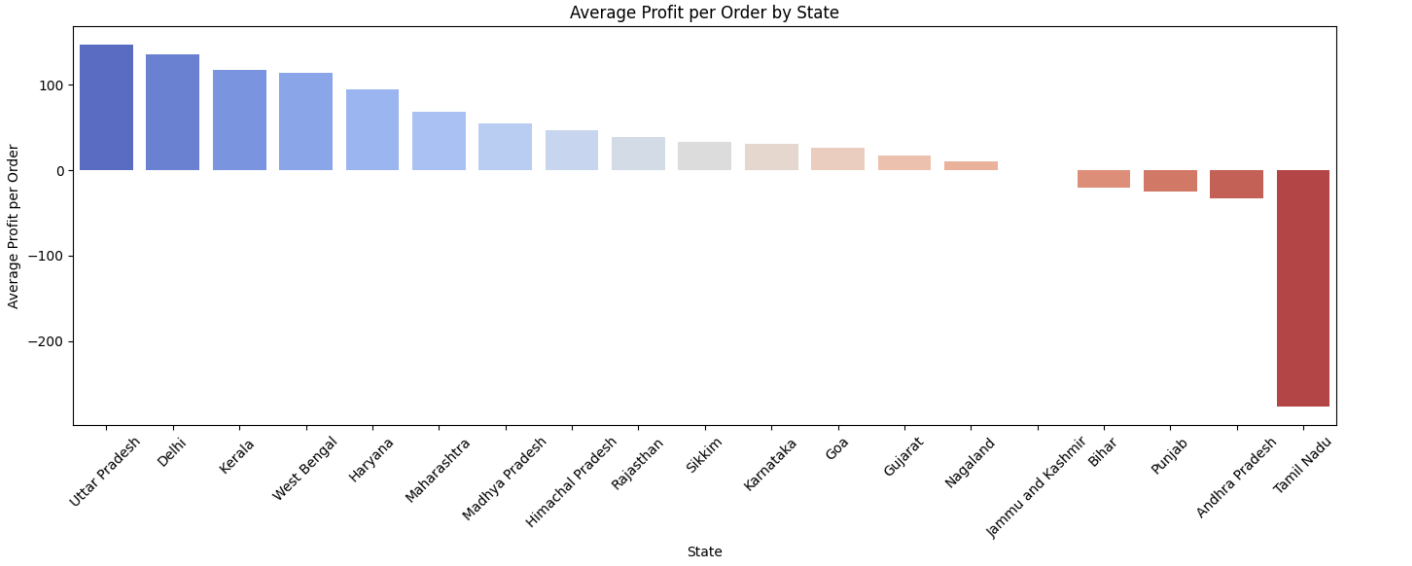
5. Final Takeaways 🚀

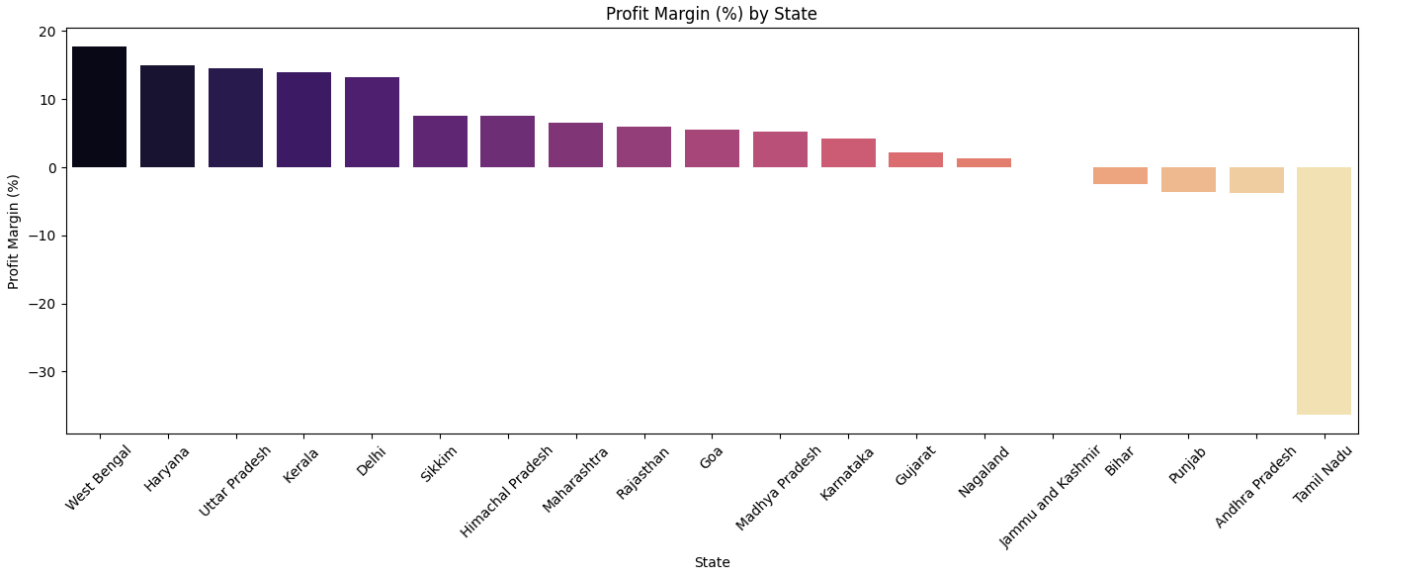
- **Maharashtra and Madhya Pradesh** are top-performing states, with strong sales and profitability.
- **Punjab and Gujarat** need urgent improvements, especially in pricing, operational efficiency, and product strategies.
- **Rajasthan** shows moderate performance, with room for improvement in pricing optimization.
- Regional disparities indicate the need for tailored business strategies, focusing on improving logistics, pricing models, and customer demand analysis.

```python
# Identify the top 5 states by order count using the orders dataset
top_states = orders['State'].value_counts().head(5).index.tolist()
print("Top 5 states with highest order count:", top_states)

# Merge the orders and order_details datasets on 'Order ID'
merged_df = pd.merge(orders, order_details, on='Order ID', how='inner')

# Filter the merged dataset to include only orders from the top 5 states
top_states_df = merged_df[merged_df['State'].isin(top_states)]

# Group by State to calculate total sales and average profit
state_metrics = top_states_df.groupby('State').agg(
    total_sales=('Amount', 'sum'),
    avg_profit=('Profit', 'mean')
).reset_index()

# Display the results
print("\nTotal Sales and Average Profit for Top 5 States:")
print(state_metrics)
```

```
Top 5 states with highest order count: ['Madhya Pradesh', 'Maharashtra', 'Rajasthan', 'Gujarat', 'Punjab']

Total Sales and Average Profit for Top 5 States:
           State  total_sales  avg_profit
0        Gujarat      21058.0    5.344828
1  Madhya Pradesh     105140.0   16.326471
2    Maharashtra      95348.0   21.296552
3         Punjab      16786.0  -10.150000
4      Rajasthan      21149.0   16.986486
```


Total Sales by State
Total Sales
100000
80000
60000
40000
20000
0
Madhya Pradesh
Maharashtra
Delhi
Uttar Pradesh
Rajasthan
Gujarat
Punjab
Karnataka
West Bengal
Kerala
Andhra Pradesh
Bihar
Nagaland
Jammu and Kashmir
Haryana
Himachal Pradesh
Goa
Tamil Nadu
Sikkim
State


Average Profit per Order by State
Average Profit per Order
100
0
−100
−200
Uttar Pradesh
Delhi
Kerala
West Bengal
Haryana
Maharashtra
Madhya Pradesh
Himachal Pradesh
Rajasthan
Sikkim
Karnataka
Goa
Gujarat
Nagaland
Jammu and Kashmir
Bihar
Punjab
Andhra Pradesh
Tamil Nadu
State


Profit Margin (%) by State
Profit Margin (%)
20
10
0
−10
−20
−30
West Bengal
Haryana
Uttar Pradesh
Kerala
Delhi
Sikkim
Himachal Pradesh
Maharashtra
Rajasthan
Goa
Madhya Pradesh
Karnataka
Gujarat
Nagaland
Jammu and Kashmir
Bihar
Punjab
Andhra Pradesh
Tamil Nadu
State

Question 2 : (10 marks) App Exploration: Explore the features and user experience of the Jar app. Highlight five things you found particularly e ective and user-friendly. Additionally, identify five areas where improvements could be made, providing your reasoning for each suggestion.

Jar App: Features & Areas for Improvement

📌 **Overview**

The **Jar app** is a **daily gold savings platform** designed to facilitate **micro-investments** by automatically rounding up users' online transactions and investing the spare change into **digital gold**. This innovative approach helps inculcate a **habit of regular savings** among users.

✅ **Five Effective and User-Friendly Features**

1 Automated Micro-Savings

🔹 **How It Works:** Jar seamlessly integrates with users' transactions, rounding up each purchase to the nearest **₹10** and investing the difference into **24-karat digital gold**.

🔹 **Example:** If a user spends **₹27**, the app rounds it up to **₹30**, allocating **₹3** to gold savings.

🔹 **Why It's Beneficial:** This feature **simplifies** the savings process, enabling users to **accumulate wealth effortlessly** without needing significant upfront investments.

2 Simple and Quick Setup

🔹 **User Experience:** The app has an **intuitive design**, ensuring easy navigation.

🔹 **Speed:** The **onboarding process** is quick, allowing new users to **set up their accounts and start saving in just 45 seconds**.

🔹 **Impact:** A streamlined interface **enhances user engagement**, making financial management **accessible** even to those less familiar with digital platforms.

3 Real-Time Gold Price Updates

🔹 **Feature:** Jar provides **live gold price updates** through its **'Gold Price Tool'**, refreshing **every minute**.

🔹 **Advantage:** Users can **monitor real-time prices** and make **informed investment decisions** aligned with market trends.

4 Educational Content

🔹 **What It Offers:** The app features **financial tips and educational content** covering topics such as:

- Effective saving strategies 💰
- Understanding digital gold investments 🏅
  - 🔹 **Benefit:** Educated users can **make smarter financial decisions**, improving their overall **investment strategies**.

5 Faster & More Responsive App Performance

🔹 **Enhancement:** Recent updates **optimized** the app's performance, reducing **startup times by 45%**.

🔹 **User Impact:** A **smoother and faster experience** minimizes frustration, making users more likely to **continue using** the platform.

⚠️ **Five Areas for Improvement**

1 Withdrawal Process

❌ **Issue:**

- Users experience a **mandatory 24-hour waiting period** when withdrawing funds.
- There are **differences between buying and selling prices of digital gold**, leading to **potential losses**.
  💡 **Suggestion:**
- Implement a **more transparent and expedited withdrawal process**.
- Clearly communicate any **fees or price differences** to build **user trust**.

2 Customer Support

❌ **Issue:**

- Many users report **slow or unresponsive customer support**, especially during the **KYC (Know Your Customer) process**.
  💡 **Suggestion:**
- Improve **customer support availability** with:
  ✅ **Live chat options** for instant assistance.
  ✅ **Faster email and ticket responses** to resolve queries efficiently.

3 App Stability & Technical Issues

❌ **Issue:**

- Some users experience **frequent sign-outs**, **auto-pay malfunctions**, and **app crashes**.
  💡 **Suggestion:**
- Release **regular updates** focusing on:
  ✅ **Bug fixes** to prevent crashes.
  ✅ **System stability improvements** for a seamless user experience.

4 Discoverability of Features

❌ **Issue:**

- Users may find it **difficult to locate or use all available features**.
  💡 **Suggestion:**
- Introduce:
  ✅ **Guided tutorials** to help users explore features.
  ✅ **Tooltips & an in-app help section** for easy navigation.

5 Competitive Pricing Transparency

❌ **Issue:**

- Users have raised concerns about **price differences** when buying and selling digital gold.
- Some feel **gold prices in the app are higher** compared to other marketplaces.
  💡 **Suggestion:**
- Clearly **display pricing structures**, including:
  ✅ **Fees, margins, and price variations** between buying & selling.
  ✅ Offering **competitive rates** to boost **user confidence** and trust.

🚀 **Final Takeaways**

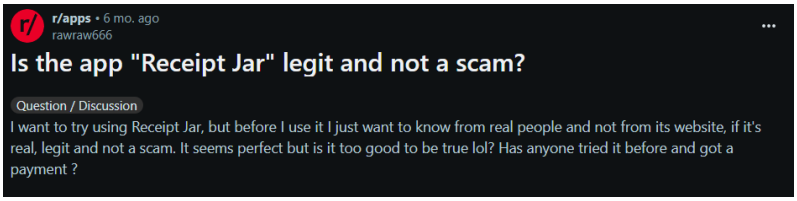
✅ **Strengths:**

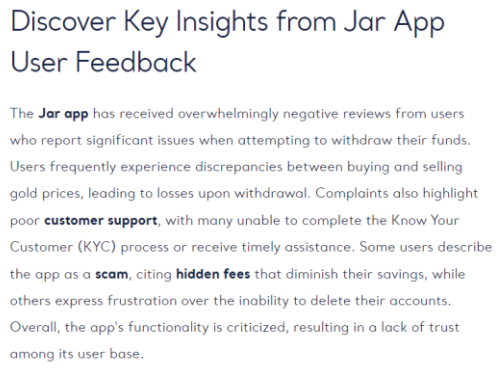
- Jar excels in **automated micro-savings, real-time pricing, user-friendly design, and educational content**.
- Recent **performance optimizations** have improved **user experience** significantly.

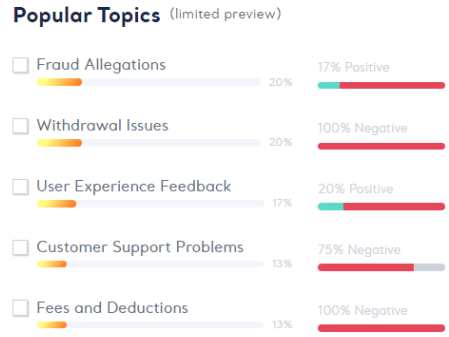
⚠️ **Areas Needing Improvement:**

- **Withdrawal delays, unresponsive support, app stability issues, feature discoverability, and pricing transparency** are key concerns.
- Addressing these issues will **enhance trust, boost engagement, and solidify Jar's position** as a leading gold-saving platform.

By making these improvements, **Jar can offer an even better user experience**, fostering **greater trust and long-term engagement** among its growing customer base. 🏅💰







These were some of the conclusions that I could make from the user reviews online there are more but these are enough for now

Question 3: (10 marks) 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 o erings 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.

Expanding Business Opportunities for Jar

1. Broadened Investment Options

🔹 **Opportunity:**
▫ In addition to digital gold, Jar could enable investments in other asset classes such as silver, mutual funds, recurring deposits, or even fractional shares of stocks.

🔹 **How It Leverages Strengths:**
✅ **Automation:** The existing round-up and micro-investment mechanism can be adapted to automatically invest in diversified portfolios.
✅ **User Trust:** With its established credibility in digital gold, users may be more open to exploring other investment vehicles if the transition is seamless.

🔹 **Potential Impact:**
📈 Broadening investment options would not only diversify users' portfolios but also attract a wider audience seeking multiple investment solutions in one platform.

---

2. Robo-Advisory Services

🔹 **Opportunity:**
▫ Jar could introduce a **robo-advisory feature** that provides **personalized, algorithm-driven investment advice**, helping users set and achieve financial goals.

🔹 **How It Leverages Strengths:**
✅ **User-Friendly Design:** A simple, interactive interface can make complex financial planning accessible to novice investors.
✅ **Automation:** Automated insights and recommendations based on user behavior and market trends can help tailor the advice to each individual's goals.

🔹 **Potential Impact:**
📊 This could deepen the app's value proposition by moving from **passive saving** to **active wealth creation** and financial planning.

---

3. Integrated Financial Management Tools

🔹 **Opportunity:**
▫ Introducing a suite of **personal finance management tools** (budget tracking, expense categorization, bill reminders) would make the Jar app a **one-stop financial hub**.

🔹 **How It Leverages Strengths:**
✅ **Automation:** Data gathered from users' spending can be used to automatically generate insights and actionable recommendations.
✅ **Established Credibility:** Users who trust Jar for their savings might be more likely to adopt additional financial management features on the same platform.

🔹 **Potential Impact:**
📌 **Enhanced engagement** through regular interactions and a holistic approach to personal finance, driving **user retention and cross-selling opportunities**.

---

4. Gamification and Community Engagement

🔹 **Opportunity:**
▫ By integrating **gamified elements**—such as savings challenges, rewards, badges, or community leaderboards—Jar can make financial planning **more engaging**.

🔹 **How It Leverages Strengths:**
✅ **User-Friendly Interface:** The intuitive design can incorporate fun and interactive elements without compromising functionality.
✅ **Automation:** Automatic tracking of savings goals and progress can fuel these gamified features, making achievements visible and rewarding.

🔹 **Potential Impact:**
🏆 **Gamification** can increase user engagement, encourage **regular usage**, and foster a **sense of community**, potentially attracting younger demographics.

---

5. Strategic Partnerships and Ecosystem Expansion

🔹 **Opportunity:**
▫ Jar can **partner with banks, insurance companies, and e-commerce platforms** to offer exclusive benefits such as **cashbacks, lower fees, or bundled financial products**.

🔹 **How It Leverages Strengths:**
✅ **Established Credibility:** Its reputation as the **first Made-in-India** micro-investment platform positions Jar as a trusted intermediary, making partnerships more appealing.
✅ **Automation & Integration:** Seamless integration with partner services can be achieved through **APIs**, allowing for **automated rewards, discounts, or investment top-ups**.

🔹 **Potential Impact:**
🌐 These **partnerships** would enhance the **overall value proposition**, increase **brand exposure**, and provide users with tangible benefits that extend beyond savings and investments.

Integrating New Services Seamlessly

Jar's success lies in its **automation, intuitive design, and trusted brand**. Here's how these can be leveraged to integrate new services:

🔹 📊 **Unified Dashboard:**
▫ A **revamped dashboard** that presents all financial data—from **micro-investments to personal finance tools and robo-advisory insights**—can make it easy for users to navigate multiple services.

🔹 🤖 **Personalized User Experience:**
▫ Using **machine learning algorithms**, Jar can offer **personalized recommendations** across all its services, ensuring that users receive tailored advice and product suggestions.

🔹 📢 **Consistent Brand Messaging:**
▫ Maintaining the **simplicity and clarity** of the current user interface while expanding features will be key. The design should emphasize **ease of use, transparency, and reliability** to keep user trust intact.

🔹 📥 **Continuous Feedback Loop:**
▫ Integrating **in-app surveys and usage analytics** will help Jar refine new offerings based on **real user behavior and feedback**, ensuring that the evolution of services aligns with user needs and expectations.

Conclusion

By **diversifying investment options, introducing robo-advisory, adding comprehensive financial management tools, incorporating gamification, and forging strategic partnerships**, Jar can significantly **enhance its value proposition**.

These opportunities not only align with its current strengths—**automation, user-friendly design, and established credibility**—but also pave the way for **deeper user engagement** and **long-term growth** in the competitive **financial ecosystem**. 🚀
