Assignment for Business Analyst Intern @Jar

Name: Avinash Soy
Email: avisoy2002@gmail.com
Contact: +919752041517, +918817844981

The whole assignment is done in Python
Assignment Link(Google Colab):-
https://colab.research.google.com/drive/1PjycOujfdIaewnvtMty2mqTALCL0dCuH?usp=sharing

Question 1 : (30 marks)

Sales Analysis:

Part 1: Sales and Profitability Analysis

• 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.

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

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

```
#Import the required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly
```

```
# Load datasets
orders_df = pd.read_csv("/content/drive/MyDrive/List_of_Orders_55FFC79CF8.csv").dropna() #drop the null values
target_df = pd.read_csv("/content/drive/MyDrive/Sales_target_DD2E9B96A0.csv").dropna() #drop the null values
details_df = pd.read_csv("/content/drive/MyDrive/Order_Details_19795F61CF.csv").dropna()
```

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
...	...	...	...	...	...
495	B-26096	28-03-2019	Atharv	West Bengal	Kolkata
496	B-26097	28-03-2019	Vini	Karnataka	Bangalore
497	B-26098	29-03-2019	Pinky	Jammu and Kashmir	Kashmir
498	B-26099	30-03-2019	Bhishm	Maharashtra	Mumbai
499	B-26100	31-03-2019	Hitika	Madhya Pradesh	Indore

500 rows × 5 columns

```
target_df #
```

	Month of Order Date	Category	Target
0	Apr-18	Furniture	10400.0
1	May-18	Furniture	10500.0
2	Jun-18	Furniture	10600.0
3	Jul-18	Furniture	10800.0
4	Aug-18	Furniture	10900.0
5	Sep-18	Furniture	11000.0
6	Oct-18	Furniture	11100.0
7	Nov-18	Furniture	11300.0
8	Dec-18	Furniture	11400.0
9	Jan-19	Furniture	11500.0
10	Feb-19	Furniture	11600.0
11	Mar-19	Furniture	11800.0
12	Apr-18	Clothing	12000.0
13	May-18	Clothing	12000.0
14	Jun-18	Clothing	12000.0
15	Jul-18	Clothing	14000.0
16	Aug-18	Clothing	14000.0
17	Sep-18	Clothing	14000.0
18	Oct-18	Clothing	16000.0

```
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
...	...	...	...	...	...	...
1495	B-26099	835.0	267.0	5	Electronics	Phones
1496	B-26099	2366.0	552.0	5	Clothing	Trousers
1497	B-26100	828.0	230.0	2	Furniture	Chairs
1498	B-26100	34.0	10.0	2	Clothing	T-shirt
1499	B-26100	72.0	16.0	2	Clothing	Shirt

1500 rows × 6 columns

```
# PART 1

merged_df = pd.merge(details_df, orders_df, on="Order ID") #merge the list of orders excelsheet with order details on the basis of order_id
merged_df
```

	Order ID	Amount	Profit	Quantity	Category	Sub-Category	Order Date	CustomerName	State	City
0	B-25601	1275.0	-1148.0	7	Furniture	Bookcases	01-04-2018	Bharat	Gujarat	Ahmedabad
1	B-25601	66.0	-12.0	5	Clothing	Stole	01-04-2018	Bharat	Gujarat	Ahmedabad
2	B-25601	8.0	-2.0	3	Clothing	Hankerchief	01-04-2018	Bharat	Gujarat	Ahmedabad
3	B-25601	80.0	-56.0	4	Electronics	Electronic Games	01-04-2018	Bharat	Gujarat	Ahmedabad
4	B-25602	168.0	-111.0	2	Electronics	Phones	01-04-2018	Pearl	Maharashtra	Pune
...	...	...	...	...	...	...	...	...	...	...
1495	B-26099	835.0	267.0	5	Electronics	Phones	30-03-2019	Bhishm	Maharashtra	Mumbai
1496	B-26099	2366.0	552.0	5	Clothing	Trousers	30-03-2019	Bhishm	Maharashtra	Mumbai
1497	B-26100	828.0	230.0	2	Furniture	Chairs	31-03-2019	Hitika	Madhya Pradesh	Indore
1498	B-26100	34.0	10.0	2	Clothing	T-shirt	31-03-2019	Hitika	Madhya Pradesh	Indore
1499	B-26100	72.0	16.0	2	Clothing	Shirt	31-03-2019	Hitika	Madhya Pradesh	Indore

1500 rows × 10 columns

```
category_sales = merged_df.groupby("Category")["Amount"].sum().reset_index() #calculate the sum of amount of each category
category_profit = merged_df.groupby("Category")["Profit"].sum().reset_index() #calculate the sum of profit of each category
category_orders = merged_df.groupby("Category")["Order ID"].nunique().reset_index() #calculate the no.of order executed for each category
```

```
category_sales
```

	Category	Amount
0	Clothing	139054.0
1	Electronics	165267.0
2	Furniture	127181.0

t steps: Generate code with category_sales | View recommended plots | New interactive sheet

```
category_profit
```

	Category	Profit
0	Clothing	11163.0
1	Electronics	10494.0
2	Furniture	2298.0

```
category_orders #No.of Orders executed in each category
```

	Category	Order ID
0	Clothing	393
1	Electronics	204
2	Furniture	186

t steps: Generate code with category_orders | View recommended plots | New interactive sheet

```
# Calculate metrics
category_analysis = category_sales.merge(category_profit, on="Category").merge(category_orders, on="Category")
#merge each category sales with their profit and calculate avg profit/order and their profit margin
category_analysis["Avg Profit per Order"] = category_analysis["Profit"] / category_analysis["Order ID"]
category_analysis["Profit Margin %"] = (category_analysis["Profit"] / category_analysis["Amount"]) * 100
```

```
category_analysis
```

	Category	Amount	Profit	Order ID	Avg Profit per Order	Profit Margin %
0	Clothing	139054.0	11163.0	393	28.404580	8.027817
1	Electronics	165267.0	10494.0	204	51.441176	6.349725
2	Furniture	127181.0	2298.0	186	12.354839	1.806874

```python
# Identify top and underperforming categories
top_category = category_analysis.sort_values("Profit Margin %", ascending=False).iloc[0]
underperforming_category = category_analysis.sort_values("Profit Margin %", ascending=True).iloc[0]
#in terms of Profit Margins, it tells category performs better or underperformed
print("Top Performing Category:")
print(top_category)
print("\nUnderperforming Category:")
print(underperforming_category)
```

```
Top Performing Category:
Category                Clothing
Amount                  139054.0
Profit                   11163.0
Order ID                     393
Avg Profit per Order    28.40458
Profit Margin %         8.027817
Name: 0, dtype: object

Underperforming Category:
Category                Furniture
Amount                   127181.0
Profit                     2298.0
Order ID                      186
Avg Profit per Order    12.354839
Profit Margin %          1.806874
Name: 2, dtype: object
```

REASONS FOR CLOTHING AS A LEADING CATEGORY

- Clothing is an affordable and convenient option for consumers.
- Typically priced lower, customers often purchase multiple clothing items in a single transaction, boosting overall sales.
- The production costs are relatively low, and seasonal promotions further enhance sales opportunities.
- Additionally, clothing items generally have higher markup percentages compared to furniture.

REASONS FOR FURNITURE UNDERPERFORMANCE

- Furniture tends to be costly, requiring significant investments, which results in fewer purchases.
- Transportation and maintenance expenses are also higher for furniture.
- Consumers tend to buy furniture less frequently, which diminishes total sales volume.
- Longer delivery and installation times can create obstacles for potential buyers.
- Economic factors, such as fluctuations in the housing market, can also affect the demand for furniture.

<u>Part 2</u>: 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.

```python
# Part 2: Target Achievement Analysis for Furniture

#Using the Sales Target dataset, calculate the percentage change in target sales for the Furniture category month-over-month.
furniture_target = target_df[target_df["Category"] == "Furniture"]
furniture_target.sort_values("Month of Order Date", inplace=True)

furniture_target["Target Change %"] = furniture_target["Target"].pct_change() * 100
furniture_target
```

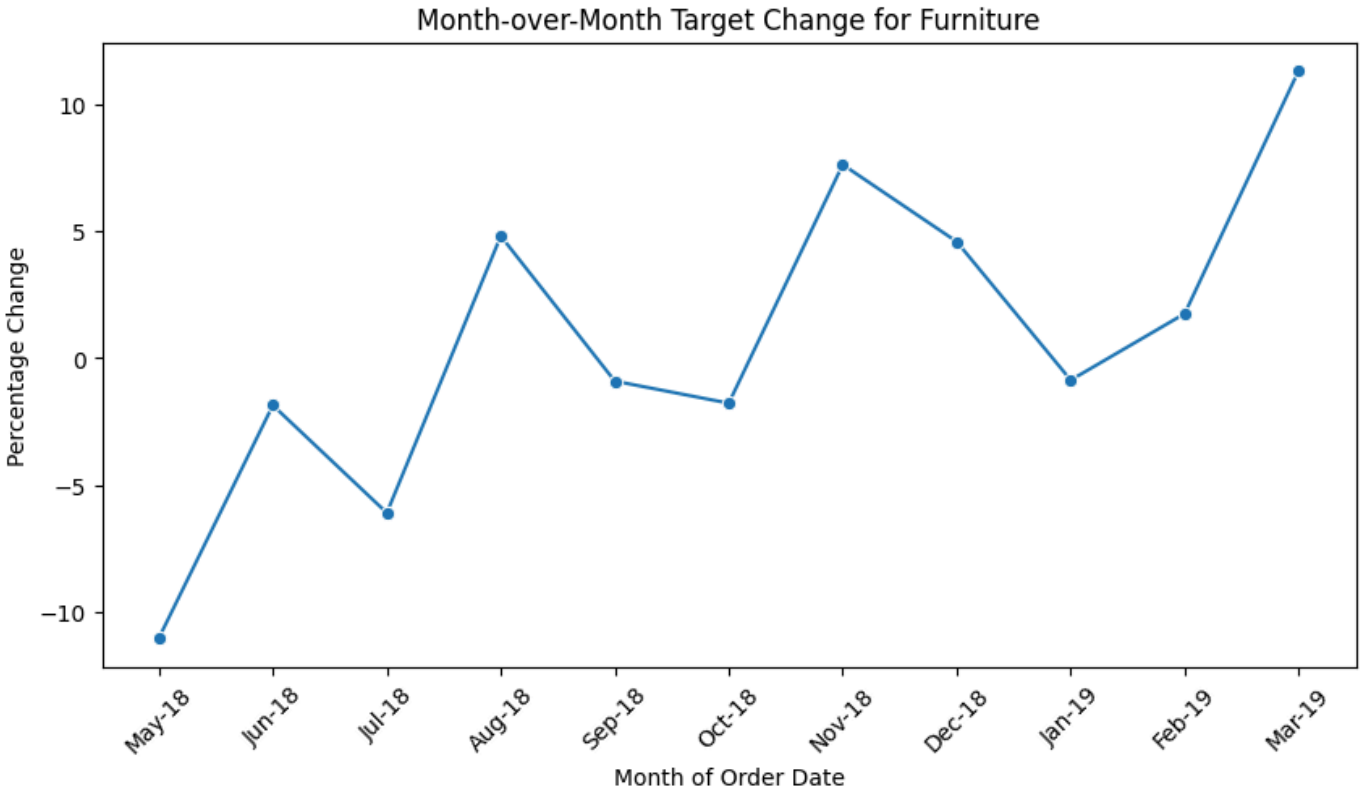
```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  furniture_target["Target Change %"] = furniture_target["Target"].pct_change() * 100
```

	Month of Order Date	Category	Target	Target Change %
0	Apr-18	Furniture	10400.0	NaN
4	Aug-18	Furniture	10900.0	4.807692
8	Dec-18	Furniture	11400.0	4.587156
10	Feb-19	Furniture	11600.0	1.754386
9	Jan-19	Furniture	11500.0	-0.862069
3	Jul-18	Furniture	10800.0	-6.086957
2	Jun-18	Furniture	10600.0	-1.851852
11	Mar-19	Furniture	11800.0	11.320755
1	May-18	Furniture	10500.0	-11.016949
7	Nov-18	Furniture	11300.0	7.619048
6	Oct-18	Furniture	11100.0	-1.769912
5	Sep-18	Furniture	11000.0	-0.900901

```
furniture_target=furniture_target.dropna() #drop null values in furniture target
#since we calculate the target change w.r.t previous month, so 1st value will have
#target changes as null
furniture_target=furniture_target.sort_index()
furniture_target
```

	Month of Order Date	Category	Target	Target Change %
1	May-18	Furniture	10500.0	-11.016949
2	Jun-18	Furniture	10600.0	-1.851852
3	Jul-18	Furniture	10800.0	-6.086957
4	Aug-18	Furniture	10900.0	4.807692
5	Sep-18	Furniture	11000.0	-0.900901
6	Oct-18	Furniture	11100.0	-1.769912
7	Nov-18	Furniture	11300.0	7.619048
8	Dec-18	Furniture	11400.0	4.587156
9	Jan-19	Furniture	11500.0	-0.862069
10	Feb-19	Furniture	11600.0	1.754386
11	Mar-19	Furniture	11800.0	11.320755

```
# Plot trend
plt.figure(figsize=(10, 5))
sns.lineplot(x="Month of Order Date", y="Target Change %", data=furniture_target, marker="o")
plt.xticks(rotation=45)
plt.title("Month-over-Month Target Change for Furniture")
plt.ylabel("Percentage Change")
plt.show()
```



From the graph, we observe significant **month-over-month** fluctuations in the sales targets for furniture:

1. **May 2018:** There's as sharp drop of 10%, indicating a lower target possibly due to seasonality or slow demand. Also Demand of electronics items such as AC or TV due to cricket season might hamper the budget of an individual.
2. **June 2018 - August 2018:** There's a recovery, which results with a spike of 5%, suggesting an increase in sales projections.
3. **September 2018 - October 2018:** There's a Stagnation (~0%), meaning no major changes in targets.
4. **November 2018:** There's a Big increase of around ~6%, likely due to holiday season sales expectations.
5. **January 2019:** Dip towards 0%, suggesting a post-holiday slump.
6. **March 2019:** There's a sharp increase of around (~10%), indicating high expectations for furniture sales.

Strategies for Aligning Target Expectations with Trends

1. **Modify Targets According to Seasonal Trends**
- Lower targets during months that typically show weak performance, such as January and May.
- Conversely, raise targets during high-demand periods like November and March to capitalize on seasonal shopping behaviors.
2. **Recognize External Factors**
- Furniture sales can be influenced by various external elements, including economic conditions, seasonal demand for other products, housing market dynamics, and consumer spending patterns. Revise forecasts based on thorough market research and competitor insights.
3. **Enhance Inventory and Marketing Coordination**
- Ensure that sufficient inventory and promotional activities are established prior to anticipated target increases in August, November, and March. Initiate early marketing campaigns ahead of peak months to encourage pre-orders.
4. **Closely Track Monthly Performance**

- Utilize real-time data to evaluate whether targets are overly ambitious or too conservative. Make adjustments in a flexible manner instead of implementing significant changes abruptly.

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.

```
# 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.
state_orders = orders_df["State"].value_counts().nlargest(5).reset_index()
#to find state with 5 largest orders
state_orders.columns = ["State", "Order Count"]
#choose only column state and order count
regional_sales = merged_df.groupby("State")["Amount"].sum().reset_index()
#grouping state and their amount
regional_profit = merged_df.groupby("State")["Profit"].mean().reset_index()
##grouping state and their profit
regional_analysis = state_orders.merge(regional_sales, on="State").merge(regional_profit, on="State")
#merge the three information on basis of state
print("\nTop 5 States with Highest Orders:")
print(regional_analysis)

# Highlight regional disparities
low_performance_states = regional_analysis.sort_values("Profit", ascending=True).head(2)
print("\nStates with Lowest Profitability:")
print(low_performance_states)
```

```
Top 5 States with Highest Orders:
            State  Order Count    Amount     Profit
0  Madhya Pradesh          101  105140.0  16.326471
1     Maharashtra           90   95348.0  21.296552
2       Rajasthan           32   21149.0  16.986486
3         Gujarat           27   21058.0   5.344828
4          Punjab           25   16786.0 -10.150000

States with Lowest Profitability:
     State  Order Count   Amount     Profit
4   Punjab           25  16786.0 -10.150000
3  Gujarat           27  21058.0   5.344828
```

Strategies for Enhancing Performance in Punjab and Gujarat

1. **Refine Pricing Approach**
   - Punjab: The current negative profitability indicates the need for significant discounts or elevated costs. It is essential to optimize pricing structures and minimize unnecessary discounts.
   - Gujarat: Although profitability is slightly positive, it remains low. Explore options such as product bundling or implementing strategic promotional offers.
2. **Concentrate on High-Margin Products**
   - Assess the inventory for low-margin items in these regions and shift focus towards products that yield higher margins
   - Additionally, promote value-added services, such as extended warranties or premium delivery options.
3. **Streamline Logistics and Reduce Costs**

- Punjab: Elevated costs may stem from expensive logistics. Consider negotiating more favorable shipping rates or utilizing regional warehouses to cut expenses.
- Gujarat: Enhance supply chain efficiency by fostering better relationships with vendors and optimizing inventory levels.

4. **Enhance Marketing and Customer Engagement**
    - Initiate targeted marketing campaigns aimed at attracting high-value customers. Provide personalized offers that align with local shopping behaviors.
5. **Broaden Business Collaborations**
    - Engage with local enterprises for cross-promotional opportunities. Boost visibility through partnerships with local influencers or online platforms.

Question 2 : (10 marks)

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.

Five Effective and User Friendly Features:

1. **Automated Micro-Savings:** The primary function of the app, which involves rounding up transactions (for instance, a ₹92 purchase rounded to ₹100, with the additional ₹8 allocated to digital gold), is both smooth and user-friendly. This automation alleviates the cognitive burden of saving, making it particularly beneficial for individuals who find it challenging to maintain financial discipline. Furthermore, its compatibility with UPI platforms such as Paytm improves accessibility by utilizing users' established payment behaviors.
2. **Low Entry Barrier:** Investors can begin their journey with a minimum investment of just ₹10, making gold investment accessible to a wider audience. This minimal entry point is especially beneficial for newcomers or individuals with restricted disposable income, promoting empowerment and inclusivity without necessitating a large initial investment.
3. **Real-Time Gold Price Tracking:** The application offers real-time updates on gold prices, enabling users to make well-informed choices regarding their buying or selling activities. This level of transparency enhances user experience by fostering trust and empowering users to manage their investments effectively, which aligns with the app's objective of promoting financial literacy.
4. **Flexible Investment Options:** In addition to round-ups, users have the option to establish savings plans on a daily, weekly, or monthly basis, or to invest larger sums manually. This versatility accommodates various saving approaches—be it passive or active—allowing the app to align with personal preferences and improving the overall user experience.
5. **Gamified Rewards System:** The "spin a jar" feature enhances the user experience by allowing individuals to earn rewards or small gold bonuses following their investments. This element of gamification leverages principles of behavioral psychology, promoting regular engagement by transforming the act of saving into an enjoyable and rewarding activity rather than a tedious task.

Five Areas For Improvement:-

1. **Customer Support Responsiveness**
   Suggestion: Enhance the customer support with faster response times and multiple contact channels (such as option of live chat or chatbot, phone, mail, video demo etc).
   Reasoning: Users frequently report delays or difficulties in reaching support for issues like withdrawals, any urgent transaction or KYC verification. Slow responses can erode trust, especially in a financial app where timely assistance is critical for user satisfaction.
2. **Withdrawal Process Efficiency**
   Suggestion: Reduce the 24-hour withdrawal wait time and clarify associated fees upfront.
   Reasoning: The mandatory delay (e.g., funds invested Monday morning are only available

Tuesday morning) can frustrate to those users, needing quick access to money. Hidden or unexpected withdrawal fees also diminish returns, leading to perceptions of unfairness.

3. **Gold Pricing Transparency**
   Suggestion: Narrow the buy-sell price spread and align rates closer to market values.
   Reasoning: Users frequently observe that the app's purchase price is marginally elevated (for instance, ₹5050/gm compared to the market rate of ₹5000/gm), while the selling price tends to be lower than prevailing market prices. This discrepancy in pricing diminishes profitability, which may discourage long-term engagement unless users are willing to maintain their investments for extended periods, a practice that is not always feasible.
4. **App Stability and Bug Fixes**
   Suggestion: Prioritize regular updates to address crashes, sign-outs, and auto-pay glitches.
   Reasoning: Frequent application crashes and inconsistent auto-pay features can significantly hinder the user experience. For an app that depends on automation, ensuring stability is essential to uphold user trust and functionality.
5. **Educational Resources**
   Suggestion: Add in-app tutorials or guides on gold investment and financial planning.
   Reasoning: Mainly, The app is designed for users with limited financial knowledge; however, it falls short in offering comprehensive educational resources that clarify gold price changes and investment strategies. By incorporating such content, users could be better equipped to make informed decisions, which would likely improve both retention and overall satisfaction.

QUESTION 3:

New Business Opportunities for Jar

1. **Micro-Investment in Mutual Funds and Stocks**
   - Opportunity: Expand micro-investment options beyond digital gold to include mutual funds and stocks, enabling users to diversify their investment portfolios with minimal amounts.
   - Integration with Strengths: Jar’s automation feature can round up transactions and invest the spare change into low-cost index funds or fractional shares, preserving its effortless, habit-forming approach. The platform's user-friendly interface can demystify complex investment ideas through intuitive dashboards and goal-oriented prompts (e.g., “Save ₹20 daily for a vacation”). The credibility gained from its successful gold investment offerings fosters user trust in Jar's broader financial products.
   - Value Proposition: By promoting diversification, Jar can lessen dependence on gold, attracting younger, risk-savvy users. The goal could be to onboard 5 million users to mutual funds within two years, leveraging India’s 30 million stock market investors, which is just a small segment of its 900 million banked population.
   - Engagement Boost: Enhance user engagement by gamifying the investment journey with milestones (e.g., “First ₹1000 in stocks!”) and providing educational insights on market trends.
2. **Digital Lending and Credit Lines**
   - *Opportunity:* Offer small, instant credit lines (e.g., ₹1000–₹10,000) based on users’ savings and transaction history, addressing India’s credit-starved market.
   - *Integration with Strengths:* Automation can assess creditworthiness using UPI transaction data and savings patterns, delivering approvals in seconds. The

user-friendly interface can present clear repayment terms and savings-linked incentives (e.g., “Save ₹500 more to unlock a higher limit”). Jar’s credibility ensures users feel secure borrowing from a trusted savings platform.
    - *Value Proposition:* This complements savings by providing liquidity during emergencies, deepening Jar’s role as a financial companion. A milestone could be disbursing ₹100 crore in loans within 18 months, leveraging its 220,000 daily transactions.
    - *Engagement Boost:* Tie loan repayments to round-up savings, encouraging consistent app use.
3. **Personalized Financial Planning Tools**
    - *Opportunity: Introduce small, immediate credit lines ranging from ₹1,000 to ₹10,000, utilizing users' savings and transaction histories to cater to India's underserved credit market.*
    - *Integration with Strengths: Automation can evaluate creditworthiness by analyzing UPI transaction data and savings behavior, enabling rapid approval processes within seconds. A user-friendly interface can clearly outline repayment conditions and offer savings-related incentives (e.g., “Save an additional ₹500 to increase your credit limit”). Jar’s established reputation fosters a sense of security for users borrowing from a reliable savings platform.*
    - *Value Proposition: This initiative enhances savings by offering liquidity in times of need, reinforcing Jar’s position as a financial ally. A potential goal could be to distribute ₹100 crore in loans over the next 18 months, capitalizing on its 220,000 daily transactions.*
    - *Engagement Boost: Link loan repayments to round-up savings, promoting regular app engagement.*
4. **Insurance Micro-Products**
    - *Opportunity: Introduce small, immediate credit lines ranging from ₹1,000 to ₹10,000, utilizing users' savings and transaction histories to cater to India's underserved credit market.*
    - *Integration with Strengths: Automation can evaluate creditworthiness by analyzing UPI transaction data and savings behavior, enabling rapid approval processes within seconds. A user-friendly interface can clearly outline repayment conditions and offer savings-related incentives (e.g., “Save an additional ₹500 to increase your credit limit”). Jar’s established reputation fosters a sense of security for users borrowing from a reliable savings platform.*
    - *Value Proposition: This initiative enhances savings by offering liquidity in times of need, reinforcing Jar’s position as a financial ally. A potential goal could be to distribute ₹100 crore in loans over the next 18 months, capitalizing on its 220,000 daily transactions.*
    - *Engagement Boost: Link loan repayments to round-up savings, promoting regular app engagement.*

Seamless Integration and Milestones

- **Automation as the Foundation:** Each service utilizes Jar's capability to handle 220,000 transactions daily and automate financial processes, requiring minimal effort from users. For instance, investments in mutual funds or insurance premiums can seamlessly integrate with the round-up feature, ensuring uniformity across various offerings.
- **User-Centric Design as the Attraction:** Streamlined onboarding processes, such as a one-tap setup for mutual fund SIPs, along with engaging features like progress trackers,

enhance user interaction and decrease abandonment rates. Jar's casual and friendly tone can be applied to new services, making financial concepts more accessible.
- **Credibility as the Assurance:** Collaborations with well-known companies, including mutual fund providers, insurance firms, and jewelers, along with clear pricing—building on its partnership with SafeGold—strengthen Jar's trustworthiness, which is essential for expanding into competitive areas like lending and investments.

Deepening the Value Proposition

These initiatives establish Jar as a comprehensive financial ecosystem, extending its services from mere savings to include investing, borrowing, financial planning, insurance, and spending. By catering to a variety of needs—such as wealth generation through mutual funds, liquidity via lending, protection through insurance, and fulfilling aspirations with jewelry—Jar is able to foster long-term user retention and minimize churn. Its emphasis on micro-transactions promotes accessibility, while personalized recommendations driven by data, such as offering insurance options based on spending habits, improve user relevance.

Achieving Significant Milestones

- **User Growth:** Achieving the goal of 100 million users by 2025, as outlined in its Series A objectives, is attainable through a broader range of services, leveraging India's 650 million internet users and 450 million UPI users.
- **Revenue Streams:** In addition to SafeGold's 2% commission, Jar can generate income from mutual fund commissions (ranging from 1-2%), lending interest (between 10-15%), insurance collaborations, and e-commerce profits, with a target of reaching ₹500 crore in revenue by 2027.
- **Financial Inclusion Impact:** Enabling 20 million new investors or insured individuals within three years has the potential to transform Jar's position in India's fintech landscape, positioning it as a competitor to major players like PhonePe and Paytm**.**

Through a thoughtful expansion that aligns with its core strengths, Jar has the potential to transition from a specialized savings application to a groundbreaking financial platform, enhancing user engagement and reinforcing its position as a leader in India's financial landscape.