1.

import pandas as pd

sales\_data = pd.read\_csv('Walmart sales.csv')

branch\_performance = sales\_data.groupby('Branch\_ID').agg({'Sales\_Amount': 'sum', 'Revenue': 'sum'}).reset\_index()

city\_performance = sales\_data.groupby('City').agg({'Sales\_Amount': 'sum', 'Revenue': 'sum'}).reset\_index()

sales\_data['Avg\_Transaction\_Value'] = sales\_data['Sales\_Amount'] / sales\_data.groupby('Branch\_ID')['Sales\_Amount'].transform('count')

sales\_data['Conversion\_Rate'] = (sales\_data.groupby('Branch\_ID')['Sales\_Amount'].transform('count') / sales\_data.groupby('Branch\_ID')['Sales\_Amount'].transform('sum')) \* 100

branch\_performance = pd.merge(branch\_performance, sales\_data[['Branch\_ID', 'Avg\_Transaction\_Value', 'Conversion\_Rate']], on='Branch\_ID', how='left')

city\_performance = pd.merge(city\_performance, sales\_data.groupby('City').agg({'Avg\_Transaction\_Value': 'mean', 'Conversion\_Rate': 'mean'}).reset\_index(), on='City', how='left')

print("Branch Performance:")

print(branch\_performance)

print("\nCity Performance:")

print(city\_performance)

2.

import pandas as pd

sales\_data = pd.read\_csv(‘walmart sales. csv’)

branch\_item\_data = sales\_data.groupby('Branch\_ID').agg({'Sales\_Amount': 'sum', 'Number\_of\_Items': 'sum'}).reset\_index()

branch\_item\_data['Average\_Price\_Per\_Item'] = branch\_item\_data['Sales\_Amount'] / branch\_item\_data['Number\_of\_Items']

print("Average Price Per Item for Each Branch:")

print(branch\_item\_data[['Branch\_ID', 'Average\_Price\_Per\_Item']])

3.

import pandas as pd

sales\_data = pd.read\_csv('sales\_data.csv')

sales\_data['Date'] = pd.to\_datetime(sales\_data['Date'])

april\_2019\_data = sales\_data[(sales\_data['Date'].dt.year == 2019) & (sales\_data['Date'].dt.month == 4)]

grouped\_data = april\_2019\_data.groupby(['Product\_Line', 'Gender', 'Payment\_Method']).agg({'Sales\_Amount': 'sum', 'Revenue': 'sum'}).reset\_index()

march\_2019\_data = sales\_data[(sales\_data['Date'].dt.year == 2019) & (sales\_data['Date'].dt.month == 3)]

march\_2019\_performance = march\_2019\_data.groupby(['Product\_Line', 'Gender', 'Payment\_Method']).agg({'Sales\_Amount': 'sum', 'Revenue': 'sum'}).reset\_index()

march\_2019\_performance.columns = ['Product\_Line', 'Gender', 'Payment\_Method', 'March\_Sales', 'March\_Revenue']

merged\_data = pd.merge(grouped\_data, march\_2019\_performance, on=['Product\_Line', 'Gender', 'Payment\_Method'], how='left')

merged\_data['Sales\_Change'] = ((merged\_data['Sales\_Amount'] - merged\_data['March\_Sales']) / merged\_data['March\_Sales']) \* 100

merged\_data['Revenue\_Change'] = ((merged\_data['Revenue'] - merged\_data['March\_Revenue']) / merged\_data['March\_Revenue']) \* 100

focus\_areas = merged\_data[(merged\_data['Sales\_Change'] < 0) | (merged\_data['Revenue\_Change'] < 0)]

print("Focus Areas for Improvement in April 2019:")

print(focus\_areas[['Product\_Line', 'Gender', 'Payment\_Method', 'Sales\_Change', 'Revenue\_Change']])

4.

Two aspects that could be significantly improved in the Jar app are:

1. User onboarding and guidance

Issue: Many users may struggle to understand how to effectively use the app or may feel overwhelmed by its features without proper guidance.

Suggestion: Implement a comprehensive onboarding process that guides users through setting up their accounts, adding their financial information, and understanding key features.

Provide interactive tutorials or tooltips to explain how to use different functionalities within the app.

Offer personalized recommendations based on users' financial goals and spending patterns to help them get started and stay on track.

1. Enhance data visualization and insights

Issue: While the app may provide basic reports and insights, users may find it difficult to gain actionable insights from the data presented.

Suggestion: Improve data visualization capabilities to present information in a more intuitive and engaging way.

Use charts, graphs, and visual representations to illustrate spending patterns, trends, and areas where users can save or improve.

Offer customizable reports and filters so users can drill down into specific aspects of their finances that matter most to them.

Integrate machine learning or AI algorithms to provide more personalized and actionable insights tailored to each user's financial situation and goals.

5.

1. **Prominent Promotion**:
   * Increase visibility and prominence of the 'Spin to Win' feature within the app interface.
   * Display enticing graphics or animations to attract users' attention and encourage them to try the game.
   * Place the 'Spin to Win' option prominently on the app's home screen or main navigation menu.
2. **Incentives and Rewards**:
   * Offer attractive incentives or rewards for playing the game, such as discounts, coupons, or bonus points.
   * Highlight the potential prizes or rewards that users can win by spinning the wheel.
   * Implement a progressive reward system where users can unlock higher-value prizes by playing the game multiple times.
3. **Limited-Time Events**:
   * Introduce limited-time events or promotions centered around the 'Spin to Win' feature.
   * Create urgency by offering exclusive rewards or special prizes that are only available during the event period.
   * Leverage push notifications or in-app messages to notify users about ongoing events and encourage participation.
4. **Social Sharing and Referral Program**:
   * Encourage users to share their 'Spin to Win' experiences on social media platforms.
   * Implement a referral program where users can earn additional spins or rewards for inviting their friends to play the game.
   * Offer incentives for sharing the app or referring new users, such as bonus spins or exclusive rewards.
5. **Gamification and Interactivity**:
   * Enhance the gamification elements of the 'Spin to Win' game to make it more engaging and interactive.
   * Introduce new themes, animations, or sound effects to enhance the gaming experience.
   * Incorporate interactive elements such as mini-games or challenges within the spinning game to keep users entertained and motivated.
6. **Personalization and Targeting**:
   * Use data analytics to personalize the 'Spin to Win' experience based on users' preferences and behavior.
   * Tailor the prizes or rewards offered in the game to match users' interests and past interactions with the app.
   * Segment users based on demographic information or past engagement with the spinning game, and target them with customized promotions or incentives.
7. **Feedback and Iteration**:
   * Gather feedback from users who have played the game to understand their preferences and pain points.
   * Continuously iterate and improve the 'Spin to Win' feature based on user feedback and performance metrics.
   * Experiment with different approaches and monitor the impact on participation rates to identify what resonates best with your audience.