**Assignment for Business Analyst Intern @Jar**

1. **Walmart Sales Analysis**:

You have been given a data set to analyse and answer the following questions: Candidate can answer the questions by using any technical tool (Excel, Sheets, Python, etc..)

A. Analyze the performance of sales and revenue at the city and branch level ( 5 marks)

B. What is the average price of an item sold at each branch of the city (10 marks)

C. Analyze the performance of sales and revenue, Month over Month across the Product line, Gender, and Payment Method, and identify the focus areas to get better sales for April 2019. (15 marks)

**A.**

# Load the data and display the first few rows

data = pd.read\_excel(xls, sheet\_name='Walmart\_Sales')

data.head()

Result:

**Invoice ID** **Branch** **City** **Customer type** **Gender**

0 750-67-8428 A Yangon Member Female

1 226-31-3081 A Naypyitaw Normal Female

2 631-41-3108 A Yangon Normal Male

3 123-19-1176 B Yangon Member Male

4 373-73-7910 C Yangon Normal Male

**Product line** **Unit price** **Quantity** **Date** **Time**

0 Health and beauty 74.69 7 1/5/2019 13:08:00

1 Electronic accessories 15.28 5 3/8/2019 10:29:00

2 Home and lifestyle 46.33 7 3/3/2019 13:23:00

3 Health and beauty 58.22 8 1/27/2019 20:33:00

4 Sports and travel 86.31 7 2/8/2019 10:37:00

**Payment Rating**

0 Ewallet 9.1

1 Cash 9.6

2 Credit card 7.4

3 Ewallet 8.4

4 Ewallet 5.3

Here's the total sales (revenue) performance by city and branch:

* **Top-performing Branches**:
  + **Mandalay - Branch B**: $37,215.93
  + **Naypyitaw - Branch A**: $35,985.64
  + **Yangon - Branch B**: $35,193.51

**B.**

# Convert 'Date' to datetime format to facilitate time-based analysis

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

# Calculate Total Sales (Revenue) per transaction

data['Total Sales'] = data['Unit price'] \* data['Quantity']

# Group by City and Branch to analyze total sales and revenue performance

city\_branch\_sales = data.groupby(['City', 'Branch'])['Total Sales'].sum().reset\_index()

city\_branch\_sales.sort\_values(by='Total Sales', ascending=False, inplace=True)

city\_branch\_sales

Result

**City** **Branch** **Total Sales**

1 Mandalay B 37215.93

3 Naypyitaw A 35985.64

7 Yangon B 35193.51

4 Naypyitaw B 35157.75

5 Naypyitaw C 34160.14

0 Mandalay A 34130.09

6 Yangon A 33647.27

8 Yangon C 32302.43

2 Mandalay C 29794.62

**C.**

import pandas as pd

# Load data and convert date column

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

data['Month'] = data['Date'].dt.month

data['Year'] = data['Date'].dt.year

# Filter April 2019 data

april\_2019\_data = data[(data['Month'] == 4) & (data['Year'] == 2019)]

# 1. Monthly sales by Product Line

monthly\_sales\_product = data.groupby(['Year', 'Month', 'Product line'])['Total Sales'].sum().reset\_index()

# 2. Monthly sales by Gender

monthly\_sales\_gender = data.groupby(['Year', 'Month', 'Gender'])['Total Sales'].sum().reset\_index()

# 3. Monthly sales by Payment Method

monthly\_sales\_payment = data.groupby(['Year', 'Month', 'Payment'])['Total Sales'].sum().reset\_index()

# Insights for April 2019

april\_product\_sales = april\_2019\_data.groupby('Product line')['Total Sales'].sum()

april\_gender\_sales = april\_2019\_data.groupby('Gender')['Total Sales'].sum()

april\_payment\_sales = april\_2019\_data.groupby('Payment')['Total Sales'].sum()

1. **App Exploration:**

Explore the features and user experience of the Jar app. Identify two aspects that you think could be significantly improved and explain your reasoning behind each suggestion.

**Solution:**

The Jar app's features and user experience, focus on areas like **navigation flow**, **feature accessibility**, **design consistency**, and **transaction speed**.

Here are two common areas for improvement:

1. **Enhanced Savings Visualization**:
   * **Suggestion**: Improve the display of savings progress and goals. Adding customizable visuals or an interactive dashboard for users to track their goals could make the savings experience more engaging.
   * **Reasoning**: This change would empower users to feel more in control and motivated to reach their targets, making the app more appealing for consistent use.
2. **Simplified Onboarding for New Users**:
   * **Suggestion**: Streamline the initial setup and tutorial for new users. Presenting a quick, interactive tour highlighting the app's core features would make the learning curve easier.
   * **Reasoning**: Many users might be new to micro-saving platforms, so a simplified onboarding can boost user retention by helping users feel comfortable with the app quickly.
3. **Product Optimisation:**

The Jar app has an engagement feature called 'Spin to Win'. Right now, if 100 people come to the app each day, only 23 of them try out this spinning game. But, we know that people who spin are more likely to retain on the app and do transactions. Now, we want to get more people to play the game. So, the question is, how can we make sure that at least 50 people out of every 100 who visit the app each day will play 'Spin to Win'? What can we do to get more people interested in spinning the wheel?

**Solution:**

To increase engagement with the "Spin to Win" feature and aim for at least 50% daily participation, here are a few strategies that can help capture user interest and drive higher participation rates:

1. **Strategic Incentive Display**:
   * **Highlight Rewards**: Increase the visibility of potential rewards by showing what users stand to gain from spinning. Displaying the rewards dynamically as they scroll or highlighting user testimonials and recent wins on the homepage can build interest and curiosity.
   * **Reward Multiplier**: Offer a “first-time spin bonus” or increase the chances of winning for new users. For example, users who spin on consecutive days could earn additional rewards or higher win probabilities, which can encourage new users to start and returning users to keep playing.
2. **Integrate Spinning Prompts into App Workflow**:
   * **Onboarding Gamification**: For new users, integrate the "Spin to Win" as a step in their onboarding experience. For instance, they could spin to reveal their first savings goal or receive a bonus on initial transactions, making the feature feel like an essential part of the app experience.
   * **Timed Reminders and Notifications**: Use push notifications or in-app reminders to prompt users to spin the wheel at strategic times, like right after they complete a transaction or during peak user hours. A reminder such as, "Don’t forget to spin and win for more savings!" can nudge users toward the game.