Exploratory Data Analysis Project
Retail Sales

Project Title
Customer Segmentation (RFM) and Sales Performance Analysis for Strategic Growth

Objective

This project explores the relationship between customer behavior and sales trends using RFM segmentation integrated with sales performance analysis using MySQL. The goal is to uncover actionable insights that can drive smarter marketing strategies, optimize customer targeting, and ultimately enhance overall business performance.

Business Context
Summarize the fictional or real-world scenario.
Who is this for? Why does it matter? What decisions depend on this?

This project simulates a mid-sized retail company navigating a competitive and fast-evolving consumer market. With access to a growing pool of customer transactions, the organization wants to uncover deeper insights into purchasing behavior that can guide smarter business decisions.

The intended audience for this analysis includes marketing strategists, sales managers, and business analysts who are focused on optimizing customer engagement, improving campaign effectiveness, and identifying high-value customer segments.

Ultimately, decisions about promotional strategies, product placements, and retention efforts all depend on understanding who the customers are, how they shop, and what drives revenue.

Dataset Description

This project uses a fictional dataset representing a snapshot of a mid-sized retail operation, reflecting core elements of customer transactions and purchasing behavior.

- **Data source**: Kaggle
- **Dataset size**: 1,000 rows × 9 columns

Key fields:

- Transaction ID – unique identifier for each purchase
- Date – transaction date
- Customer ID – unique identifier for each customer
- Gender – customer gender
- Age – customer age
- Product Category – category of purchased product
- Quantity – number of items purchased
- Price per Unit – cost of a single item
- Total Amount – total sales

Data Cleaning & Preparation

Before analysis, the dataset was cleaned and preprocessed to ensure accuracy, consistency, and usability. Key preparation steps included:

- **Column renaming**: Replaced spaces with underscores and lower case all columns (e.g., price_per_unit → price_per_unit) for improved SQL readability.
- **Check for duplicates**: No duplicate found
- **Handling missing values**: Verified each column for nulls; none were found in critical fields like Customer ID, Transaction ID, or Total Amount.
- **Date formatting**: Date field: Changed data type from **text** to **date** for time-based analysis and RFM calculations.
- **Consistency checks**: Ensured uniform text casing in categorical variables (e.g., Gender, Product Category) to avoid grouping issues during segmentation.

Note: The dataset includes January 2024, but I focused only on the full year of 2023 to ensure complete and consistent analysis.

Exploratory Data Analysis (EDA)

1. **Sales performance and Customer Demographics Analysis**

SQL Queries

a. KPIs

Purpose: These KPIs provides instant clarity for decision-makers by highlighting key performance metrics by a glance, they can immediately understand how business is performing.

i. Total Sales

```
SELECT SUM(total_amount) AS total_sales
FROM retail_sales_staging
WHERE YEAR(date) = 2023;
```

	total_sales
▶	454470

Insight: Shows the total amount earned from confirmed transactions.

ii. Total Quantity Sold

```
SELECT SUM (quantity) AS total_quantity_sold
FROM retail_sales_staging
WHERE Year(date) = 2023;
```

	total_quantity_sold
▶	2510

Insight: Reflect the total units sold over a year

iii. Total Transaction

```
SELECT COUNT (DISTINCT transaction_id) AS total_transaction
FROM retail_sales_staging
WHERE Year(date) = 2023;
```

	avg_order_value
▶	455.38

Observation: I noticed that there is only one transaction per customer.

Insight: Shows the total unique transactions made by each customer.

iv. Average Order Value (AOV)

```
SELECT ROUND(SUM(total_amount)/(COUNT(DISTINCT transaction_id)),2) AS avg_order_value
FROM retail_sales_staging
WHERE Year(date) = 2023;
```

	avg_order_value
▶	455.38

Insight: Shows amount of money a customer spends per transaction

b. Sales Performance Analysis

i. Monthly sales trends

```
SELECT MONTH (date) AS month_num, date_format(date, '%M') AS month,
SUM (total_amount) AS total_revenue
FROM retail_sales_staging
WHERE Year(date) = 2023
GROUP BY month, month_num
ORDER BY total_revenue DESC;
```

	month_num	month	total_revenue
▶	5	May	53150
	10	October	46580
	12	December	44690
	2	February	44060
	8	August	36960
	6	June	36715
	7	July	35465
	1	January	35450
	11	November	34920
	4	April	33870
	3	March	28990
	9	September	23620

ii. Transaction frequency over months

```
SELECT MONTH(date) AS month_num, DATE_FORMAT(date,'%M') AS month, COUNT(*) AS
total_transaction
FROM retail_sales_staging
WHERE Year(date) = 2023
GROUP BY month, month_num
ORDER BY total_transaction DESC;
```

Result Grid | Filter Rows:

month_num	month	total_transaction
5	May	105
10	October	96
8	August	94
12	December	91
4	April	86
2	February	85
11	November	78
1	January	78
6	June	77
3	March	73
7	July	72
9	September	65

Insights:

Monthly sales trends

- Month of May shows highest sales which means it contributes most to the sales, more like due to seasonal demand or events.
- September has the lowest sales, potentially can improve or opportunity for growth.

Recommended Action: Introduce targeted promotion for September and set measurable goals.

Transaction frequency over months

- May has the highest number of transactions, showing high customers activity.
- September has the lowest transactions; low customer activity means low sales.

Recommended Action: Re-engage with customers during September through strategic campaigns.

Seasonal Trends

- Spring, month of May, and the holiday season, month of December, are key sales periods.
- These seasonal surges likely reflect consumer behaviors tied to events like back-to-school, summer breaks, or holiday gift-giving.

Recommended Action: Use these insights to plan inventory, staffing, and campaigns effectively.

iii. Product Category Distribution by sales

```
SELECT product_category, SUM(total_amount) AS total_revenue

FROM retail_sales_staging

WHERE Year(date) = 2023
```

GROUP BY product_category

ORDER BY total_revenue DESC;

product_category	total_sales
Electronics	156875
Clothing	155580
Beauty	142015

Customer Demographics

iv. Gender Distribution by Product category

SELECT product_category, gender, COUNT(*) AS gender_count, SUM(total_amount) AS total_sales
FROM retail_sales_staging
WHERE Year(date) = 2023
GROUP BY gender, product_category
ORDER BY product_category, total_sales DESC;

product_category	gender	gender_count	total_sales
Beauty	Female	166	74830
Beauty	Male	140	67185
Clothing	Female	174	81275
Clothing	Male	177	74305
Electronics	Male	171	80140
Electronics	Female	170	76735

% Sales by gender

SELECT gender, ROUND(SUM(total_amount) * 100/ (SELECT SUM(total_amount) FROM retail_sales_staging),2) AS gender_percentage
FROM retail_sales_staging
GROUP BY gender;

Result Grid | Filter Rows:

gender	gender_percentage
Male	48.94
Female	51.06

Segmenting customers by age and name each group to enable targeted insights.

SELECT
CASE
WHEN age BETWEEN 0 AND 20 THEN 'Teens (0-20)'
WHEN age BETWEEN 21 AND 30 THEN 'Young (21-30)'
WHEN age BETWEEN 31 AND 45 THEN 'Adult (31-45)'
WHEN age BETWEEN 46 AND 59 THEN 'Middle Age (46-59)'
ELSE 'Old (60+)'
END AS age_group
FROM retail_sales_staging;

Created view so I can use the above query for further analysis.

```
CREATE view age_seg AS (
SELECT gender, age, total_amount, transaction_id, product_category, quantity,
CASE
WHEN age BETWEEN 0 AND 20 THEN 'Teens (0-20)'
WHEN age BETWEEN 21 AND 30 THEN 'Young (21-30)'
 WHEN age BETWEEN 31 AND 45 THEN 'Adult (31-45)'
 WHEN age BETWEEN 46 AND 59 THEN 'Middle Age (46-59)'
ELSE 'Old (60+)'
END AS age_group
FROM retail_sales_staging);
```

age_group
Adult (31-45)
Young (21-30)
Middle Age (46-59)
Adult (31-45)
Young (21-30)
Adult (31-45)
Middle Age (46-59)
Young (21-30)
Old (60+)
Middle Age (46-59)

v. Product categories purchased by age group

```
SELECT age_group, product_category, COUNT(*) AS product_count
FROM age_seg
WHERE Year(date) = 2023
GROUP BY age_group, product_category
ORDER BY age_group, product_category, product_count DESC;
```

age_group	product_category	product_count
Adult (31-45)	Beauty	83
Adult (31-45)	Clothing	113
Adult (31-45)	Electronics	106
Middle Age (46-59)	Beauty	100
Middle Age (46-59)	Clothing	117
Middle Age (46-59)	Electronics	113
Old (60+)	Beauty	26
Old (60+)	Clothing	33
Old (60+)	Electronics	34
Teens (0-20)	Beauty	24
Teens (0-20)	Clothing	16
Teens (0-20)	Electronics	23
Young (21-30)	Beauty	73
Young (21-30)	Clothing	72
Young (21-30)	Electronics	65

Insights:

Product Categories

- Electronics contributes most to sales while beauty contributes least to sales.

Recommended Action: Ensure balanced inventory and tailored promotions across categories.

Gender Trends

- Female customers show a stronger preference for clothing, while clothing has balanced gender appeal.
- Male customers tend to favor electronics, indicating a focus on tech, gadgets, or utility-based products.

Recommended Action: Target beauty for females, electronics for males, and unisex campaigns for clothing.

Product Categories by Age Group:

- Electronics, clothing and beauty are most popular for middle-aged and adult customers. While young customers show interest to beauty products.

Recommended Action: Focus electronics ads on older groups, promote beauty to younger buyers, and maintain clothing inventory evenly.

Sales and Customer Insight Summary

Retail shows a good sales performance for the year 2023. After a strong peak in May, both sales transaction frequency declined in September before showing a clear recovery in October. Female customers among adult and middle-aged group have a strong preference for clothing. While male customers among adult and middle-aged group have a strong interest for electronics product. Beauty products are most popular to young female customers.

2. Customer Segmentation

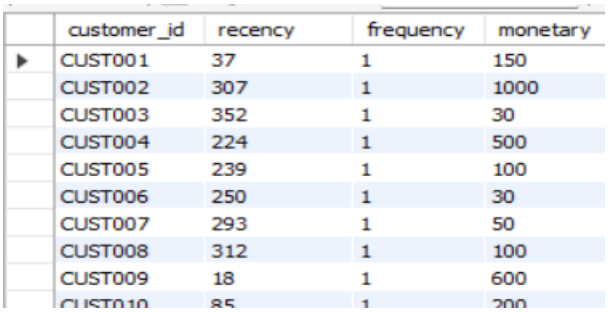
Brief description of Recency, Frequency and Monetary

- **Recency** – Days since the last purchase (lower score is better)
- **Frequency** – Total number of purchases (higher score is better)
- **Monetary** – Total amount spent (higher score is better)

In order to categorize customers, let's compute first for recency, frequency and monetary

```
SELECT customer_id,
datediff('2023-12-31', MAX (date)) AS recency_days,
COUNT (transaction_id) AS frequency,
SUM (total_amount) AS monetary
FROM retail_sales_staging
WHERE Year(date) = 2023
GROUP BY customer_id
ORDER BY frequency desc;
```

This tells how many days it's been since each customer's last purchase *as of the end of 2023*. In that way, my analysis stays consistent and accurate for that historical window.
When calculating **Recency** for RFM analysis, it's all about determining how recently each customer made a purchase **relative to your dataset's most recent date**, *not* today's real-world date.



customer_id	recency	frequency	monetary
CUST001	37	1	150
CUST002	307	1	1000
CUST003	352	1	30
CUST004	224	1	500
CUST005	239	1	100
CUST006	250	1	30
CUST007	293	1	50
CUST008	312	1	100
CUST009	18	1	600
CUST010	85	1	200

-Added RFM columns

```
ALTER TABLE retail_sales_staging
ADD COLUMN recency int,
ADD COLUMN frequency int,
ADD COLUMN frequency int;
```

-Inserted the data to each column

```
UPDATE retail_sales_staging AS r
JOIN (
SELECT customer_id,
MAX(date) AS last_purchase
FROM retail_sales_staging
WHERE Year(date) = 2023
GROUP BY customer_id
) AS recent ON r.customer_id = recent.customer_id
SET r.recency = DATEDIFF('2023-12-31', recent.last_purchase);

UPDATE retail_sales_staging AS f
JOIN (
SELECT
customer_id,
 COUNT (transaction_id) AS frequency
 FROM retail_sales_staging
WHERE Year(date) = 2023
 GROUP BY customer_id
) AS freq ON f.customer_id = freq.customer_id
SET f.frequency = freq.frequency;

UPDATE retail_sales_staging AS m
```

```
JOIN (SELECT
customer_id,
SUM (total_amount) AS monetary
FROM retail_sales_staging
WHERE Year(date) = 2023
GROUP BY customer_id
) AS mon ON m.customer_id = mon.customer_id
SET m.monetary = mon.monetary;
```

-Based on the computed RFM, I assigned RFM scores for each customer.

```
SELECT customer_id,
-- The more recent the transaction (lower number of days), the higher the recency score.
CASE
WHEN recency <= 30 THEN 5
WHEN recency <= 100 THEN 4
WHEN recency <= 190 THEN 3
WHEN recency <= 380 THEN 2
ELSE 1
END AS r_score,
-- The higher the value of frequency and monetary, the higher the score.
CASE
WHEN frequency >= 10 THEN 5
WHEN frequency >= 7 THEN 4
WHEN frequency >= 5 THEN 3
WHEN frequency >= 3 THEN 2
ELSE 1
END AS f_score,
CASE
WHEN monetary >= 1800 THEN 5
WHEN monetary >= 1350 THEN 4
WHEN monetary >= 900 THEN 3
WHEN monetary >= 450 THEN 2
ELSE 1
END AS m_score
FROM retail_sales_staging
WHERE Year(date) = 2023;
```

r_score	f_score	m_score
5	1	1
4	1	3
4	1	1
5	1	2
5	1	1
5	1	1
4	1	1
4	1	1
5	1	2

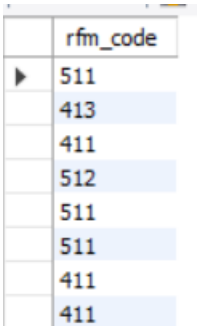
-Adding RFM score columns

```
ALTER TABLE retail_sales_staging
ADD column r_score int,
ADD column f_score int,
ADD column m_score int;
```

-Inserting data to each column

```
UPDATE retail_sales_staging
SET r_score = (SELECT
CASE
WHEN recency <= 30 THEN 5
WHEN recency <= 100 THEN 4
WHEN recency <= 190 THEN 3
WHEN recency <= 380 THEN 2
ELSE 1
END AS r_score);

UPDATE retail_sales_staging
SET f_score = (SELECT
CASE
WHEN frequency >= 10 THEN 5
WHEN frequency >= 7 THEN 4
WHEN frequency >= 5 THEN 3
WHEN frequency >= 3 THEN 2
ELSE 1
END AS f_score);

UPDATE retail_sales_staging
SET m_score = (SELECT
CASE
WHEN monetary >= 1800 THEN 5
WHEN monetary >= 1350 THEN 4
WHEN monetary >= 900 THEN 3
```

```
WHEN monetary >= 450 THEN 2
ELSE 1
END AS m_score);

-Combining the scores to create a RFM code

SELECT CONCAT (r_score, f_score, m_score) AS rfm_code
FROM retail_sales_staging;
```



rfm_code
511
413
411
512
511
511
411
411

-Adding column and insert rfm codes

-- At first, I stored the data type as int, after searching, it should be a string (varchar or char) to use the rfm_code for labeling the segments.
-- Make sure your rfm_code is stored as a string (like this '532') so you can use LIKE patterns to classify customer types flexibly.

```
ALTER TABLE retail_sales_staging
ADD column rfm_code int;

UPDATE retail_sales_staging
SET rfm_code = CONCAT (r_score, f_score, m_score);
```

Grouping customers into descriptive segments based on recency, frequency and monetary definitions and scores:

a) Champions: High Recency, Frequency, and Monetary
b) Loyal Customers: Frequent buyers with consistent value
c) At Risk: Haven't purchased in a while
d) Big Spenders: High spenders
e) Low engagement: Low activity across the board, long-term nurturing required.

Then, assign scores from 1 (lowest) to 5 (highest) for each metric. Combine the three scores into a single 3-digit number (e.g., **555** is a top customer, **111** is the least engaged).

RFM Segment	**Customer Label**
555	Champions

5xx	Loyal Customers
x5x	Big Spenders
2xx	At Risk
Others	Low Engagement

```
-- Grouping customers into descriptive segments
SELECT customer_id,
  CASE
WHEN rfm_code = '555' THEN 'Champions'
WHEN rfm_code LIKE '5__' AND rfm_code != '555' THEN 'Loyal Customers'
WHEN rfm_code LIKE '__5' THEN 'Big Spenders'
WHEN rfm_code LIKE '2__' THEN 'At Risk'
ELSE 'Low Engagement'
END AS segment_label
FROM retail_sales_staging;

-- Adding customer segment into a new column
ALTER TABLE retail_sales_staging
ADD column customer_segment VARCHAR (50);

 UPDATE retail_sales_staging
 SET customer_segment =  (SELECT
 CASE
   WHEN rfm_code = '555' THEN 'Champions'
   WHEN rfm_code LIKE '5__' AND rfm_code != '555' THEN 'Loyal Customers'
   WHEN rfm_code LIKE '__5' THEN 'Big Spenders'
   WHEN rfm_code LIKE '2__' THEN 'At Risk'
   ELSE 'Low Engagement'
 END AS customer_segment);
```

customer_segment
Low Engagement
At Risk
Low Engagement
At Risk
At Risk
At Risk
At Risk
At Risk
Low Engagement
Low Engagement
At Risk
Low Engagement
Low Engagement

i. Customers Segment (as percent)

```
SELECT customer_segment, ROUND (COUNT (customer_id) * 100 / (SELECT COUNT (customer_id) FROM
retail_sales_staging),2) AS percent_segment
FROM retail_sales_staging
GROUP BY customer_segment;
```

customer_segment	percent_segment
Low Engagement	46.00
At Risk	48.90
Big Spenders	4.90
Loyal Customers	0.20

ii. Total Sales by Segments

```
SELECT DISTINCT (customer_segment), SUM (total_amount) AS total_sales
FROM retail_sales_staging
GROUP BY customer_segment
ORDER BY total_sales DESC;
```

	customer_segment	total_sales
▶	At Risk	199715
	Low Engagement	158285
	Big Spenders	98000

iii. Average RFM Scores by Segment

```
SELECT
  customer_segment,
  AVG(r_score) AS avg_recency_score,
  AVG(f_score) AS avg_frequency_score,
  AVG(m_score) AS avg_monetary_score
FROM retail_sales_staging
GROUP BY customer_segment;
```

customer_segment	avg_recency_score	avg_frequency_score	avg_monetary_score
Low Engagement	3.2435	1.0000	1.5804
At Risk	2.0000	1.0000	1.6053
Loyal Customers	5.0000	1.0000	2.5000
Big Spenders	2.5918	1.0000	5.0000

Insights from RFM Analysis

1. **Customer Segments**:
    - The distribution indicates higher count of at-risk and low engagement, while lower count of loyal customers.

- The at-risk segment contributes more to sales despite of showing signs of declining engagement.
- Low RFM scores signal opportunities to improve through targeted reactivation efforts.

Recommended Actions:

- **Retention Focus**: Prioritize reactivation campaigns for at-risk customers, especially those high monetary value from the past purchases.
- **Value Growth**: Introduce loyalty programs, exclusive discounts, or upsell strategies for low-engagement customers. The goal is to increase their frequency and spend, moving them toward more valuable segments.
- **Customer Experience**: Identify potential reasons for inactivity through feedback mechanisms like short surveys or post-exit polls.

RFM Analysis Summary

Most customers fall into the at-risk and low engagement groups, while loyal customers are fewer. Even though at-risk customers aren't purchase as much, they still make up a big part of the sales, so it's worth focusing on keeping them. To grow value, we can re-engage inactive customers with discounts, rewards, or surveys to understand what's stopping them from buying again.

Note: Exported the data into excel for later visualization

Kindly check my tableau project for this dataset. Thank you!