CASE STUDY: "Sales Performance and Customer Segmentation (RFM) Analysis for Strategic Growth"

Introduction

In today's fast-moving retail world, understanding how customers behave is key to growing sales and boosting performance. This case study demonstrates how RFM-based customer segmentation, combined with sales performance metrics, reveal customer behavior and guide smarter business decisions.

Using a fictional retail dataset, I analyze transaction data to identify high-value customers, reveal spending habits, and assess their influence on overall business performance. The goal is to translate raw data into actionable strategies for improving customer targeting, retention, and marketing efficiency.

Business Challenge

A mid-sized retail business wants to boost revenue by understanding customer purchasing patterns and identifying its most valuable shoppers. The leadership team suspects that marketing efforts are too broad, and they're missing opportunities to personalize outreach and improve retention.

The goal of this analysis is to uncover seasonal trends and segment customers based on Recency, Frequency, and Monetary (RFM) value. The aim is to evaluate how each customer segment contributes to overall sales performance, guiding smarter decisions in marketing, inventory management, and customer engagement strategies.

The analysis will follow the 6 phases of the Data Analysis process: Ask, Prepare, Process, Analyze, Share and Act (APPASA).

ASK

Key Questions

1. Sales Performance and Customer Demographics Analysis
    i. Which month exhibit peak sales performance, and what key factors contribute to those peaks?
    ii. Which months have higher or lower customer purchase activity?
    iii. How does customer demographics such as age and gender influence spending and product preferences?
    iv. Which product contributes the most to revenue?

2. Customer Segmentation (RFM Analysis)
    i. What percentage of customers belong to the high-value segment? (e.g., Champions, Loyal Customers)?
    ii. Which RFM segments contribute the most to overall revenue?
    iii. How can we re-engage at-risk or low engagement customers?

PREPARE

This dataset gives a quick look at a fictional retail setting, showing key details about how the business runs and how customers behave.

Data Overview

- Data source: Kaggle.com
- Dataset: retail sales
- Dataset size: 1,000 rows × 9 columns
- Unique values: 1000 transaction ID

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

Tools used

- MySQL: data cleaning, RFM scoring, JOINS, and view creation

- Excel: preliminary preprocessing and exploratory checks (pivot table)

Created a database named retail_sales and imported the retail dataset to the database.
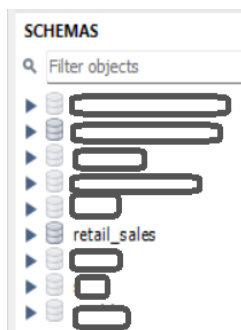
Created and used a staging table to keep the original data untouched while cleaning and analyzing it.

SQL Query:

```
CREATE TABLE retail_sales_staging
SELECT * FROM retail_sales_dataset;
```

View staging table

SQL Query:

```sql
SELECT * FROM retail_sales_staging;
```

Output:

First 15 rows

Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
6	2023-04-25	CUST006	Female	45	Beauty	1	30	30
7	2023-03-13	CUST007	Male	46	Clothing	2	25	50
8	2023-02-22	CUST008	Male	30	Electronics	4	25	100
9	2023-12-13	CUST009	Male	63	Electronics	2	300	600
10	2023-10-07	CUST010	Female	52	Clothing	4	50	200
11	2023-02-14	CUST011	Male	23	Clothing	2	50	100
12	2023-10-30	CUST012	Male	35	Beauty	3	25	75
13	2023-08-05	CUST013	Male	22	Electronics	3	500	1500
14	2023-01-17	CUST014	Male	64	Clothing	4	30	120
15	2023-01-16	CUST015	Female	42	Electronics	4	500	2000

Checking if the data is clean. If not, I will clean it first before exploring the dataset.

```sql
-- For SQL readabilty I will replace those column names with spaces in between with underscore and lower case each column.
ALTER TABLE retail_sales_staging
RENAME COLUMN `Transaction ID` TO `transaction_id`,
RENAME COLUMN `Date` TO `date`,
RENAME COLUMN `Customer ID` TO `customer_id`,
RENAME COLUMN `Gender` TO `gender`,
RENAME COLUMN `Age` TO `age`,
RENAME COLUMN `Product Category` TO `product_category`,
RENAME COLUMN `Quantity` TO `quantity`,
RENAME COLUMN `Price per Unit` TO `price_per_unit`,
RENAME COLUMN `Total Amount` TO `total_amount`;
```

Output:

Table: retail_sales_staging

Columns:

transaction_id	int
date	text
customer_id	text
gender	text
age	int
product_category	text
quantity	int
price_per_unit	int
total_amount	int

PROCESS

In data cleaning, I followed some steps:

a. Removed duplicates

SQL Query:

```sql
SELECT transaction_id, date, customer_id, gender, age, product_category, quantity, price_per_unit, total_amount, COUNT(*)
FROM retail_sales_staging
GROUP BY transaction_id, date, customer_id, gender, age, product_category, quantity, price_per_unit, total_amount
HAVING COUNT(*) > 1;
```

Output:

Number of duplicates: 0 row(s) returned

b. Identified null in each column

SQL Query:

```sql
SELECT * FROM retail_sales_staging
WHERE transaction_id IS NULL OR date IS NULL OR customer_id IS NULL OR gender IS NULL OR age IS NULL OR product_category IS NULL
OR quantity IS NULL OR price_per_unit IS NULL OR  total_amount IS NULL;
```

Output:

Identifying Nulls: 0 row(s) returned; No nulls found

c. Standardizing data types

Changed data type from text to date

SQL Query:

```sql
ALTER TABLE retail_sales_staging
MODIFY `date` DATE;
```

Output:

```
Columns:
    transaction_id      int
    date                date
    customer_id         text
    gender              text
    age                 int
    product_category    text
    quantity            int
    price_per_unit      int
    total_amount        int
```

Changed data type from text to VARCHAR

```sql
ALTER TABLE retail_sales_staging
MODIFY customer_id VARCHAR(50);
```

Output:

Table: retail_sales_staging

Columns:
transaction_id	int
date	date
customer_id	varchar(50)
gender	text
age	int
product_category	text
quantity	int
price_per_unit	int
total amount	int

After verifying that the data is clean by removing duplicates, identifying nulls and standardizing formats, I proceeded to the analysis.

ANALYZE

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

1. **Sales performance and Customer Demographics Analysis (EDA)**

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.

a. **What is the total sales amount for the year 2023?**

SQL Query:

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

Output:

total_sales
454470

b. How many units were sold overall?

SQL Query:

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

Output:

	total_quantity_sold
▶	2510

c. How many transactions were made?

SQL Query:

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

Output:

	total_transaction
▶	998

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

d. What is the average order value?

SQL Query:

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

Output:

	avg_order_value
▶	455.38

Sales Performance Analysis

Monthly sales trends

Purpose: The purpose of analyzing sales trends over a year is to identify patterns and seasonal variations in sales performance. This provides insights into:
- High-demand months where sales peak, such as holiday seasons or promotional periods.
- Low-demand months that may require strategic interventions to boost sales.

a. **Which month exhibit peak sales performance?**

SQL Query:

```sql
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;
```

Output:

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

Transaction Frequency Over Months

Purpose: The purpose of analyzing transaction frequency over a year is to understand how often transactions occur during specific months. By examining the distribution of transactions, businesses can:

- Identify high-activity months where transactions peak, indicating increased customer engagement.
- Detect low-activity months that may require targeted marketing or promotional efforts.

b. **Which months have higher or lower customer purchase activity?**

SQL Query:

```sql
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;
```

Output:

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
6	June	77
1	January	76
3	March	73
7	July	72
9	September	65

Product Category by sales

Purpose: The purpose of this is to analyze how much revenue each product category brings in.

- Identify which categories are driving the most sales.
- Helps to spot top-performing items and make decisions about inventory, marketing, or promotions.

c. **Which product contributes the most to revenue?**

SQL Query:

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

Output:

product_category	total_sales
Electronics	156875
Clothing	155580
Beauty	142015

Customer Demographics

Segmentation of Customers by Age
Grouped customers by age ranges and assigned labels to each group to better understand behavior and create focused insights.

SQL Query:

```sql
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 to store selected columns for further analysis.

SQL Query:

```sql
CREATE view age_segmentation AS
(
SELECT gender, age, total_amount, date, 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 Segmentation:

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)

Product categories purchased by age group

Purpose: To see what types of products each age group buys, so businesses can offer better choices, send the right promotions, and improve customer satisfaction.

   a. **How does customer demographics such as age influence spending and product preferences?**

   SQL Query:

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

   Output:

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	95
Middle Age (46-59)	Clothing	109
Middle Age (46-59)	Electronics	104
Old (60+)	Beauty	31
Old (60+)	Clothing	41
Old (60+)	Electronics	43
Teens (0-20)	Beauty	24
Teens (0-20)	Clothing	16
Teens (0-20)	Electronics	23
Young (21-30)	Beauty	73

Gender Distribution by Product category

Purpose: The purpose of this is to analyze how different genders contribute to sales across various product categories.

   b. **How does customer demographics such gender influence spending and product preferences?**

   SQL Query:

```
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;
```

Output:

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

2. Customer Segmentation (RFM)

Purpose: Helps businesses know which customers buy often, spend more, or might stop buying so they can improve marketing and service. By segmenting customers through RFM scores, business can:

- Tailor marketing strategies to match each group's preferences and needs
- Personalize product recommendations and promotions to increase engagement
- Identify high-value segments like loyal customers or big spenders
- Spot customers who are at risk of churning and create strategies to retain them

Brief description of Recency, Frequency and Monetary (RFM)

- 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)

By combing RFM scores, I will classify customers into actionable segments like:
- Champions – Recent, frequent, and high-value buyers
- Loyal Customers – Consistent shoppers with strong history
- Big Spenders – Customers who spend a lot
- At-Risk – Used to buy, but activity is dropping
- Low Engagement – Rarely shop and spend little

After calculating RFM scores, I created customer segments labeled 'Champions', 'Loyal Customers', 'Big Spenders', 'At Risk', and 'Low Engagement'. Then, I analyzed how customers in each segment behaved and what they tended to buy.

Below is the step-by-step process:

1. RFM Calculation (recency, frequency, monetary)

Used CTE and CROSS JOIN to calculate the recency

SQL Query:

```sql
WITH max_date AS (
    SELECT MAX(date) AS latest_date
    FROM retail_sales_staging
    WHERE YEAR(date) = 2023
)
SELECT
    customer_id,
    DATEDIFF(m.latest_date, MAX(r.date)) AS recency_days,
    COUNT(r.transaction_id) AS frequency,
    SUM(r.total_amount) AS monetary
FROM retail_sales_staging r
CROSS JOIN max_date m
WHERE YEAR(r.date) = 2023
GROUP BY customer_id, m.latest_date;
```

Output:

customer_id	recency_days	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

Created new columns to store the RFM values used for scoring and segmentation.

SQL Query:

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

Inserted the data to each column

```sql
WITH max_date AS (
    SELECT MAX(date) AS latest_date
    FROM retail_sales_staging
    WHERE YEAR(date) = 2023
),
rfm_calc AS (
    SELECT
        customer_id,
        DATEDIFF(m.latest_date, MAX(r.date)) AS recency_days,
        COUNT(r.transaction_id) AS frequency,
        SUM(r.total_amount) AS monetary
    FROM retail_sales_staging r
    CROSS JOIN max_date m
    WHERE YEAR(r.date) = 2023
    GROUP BY customer_id, m.latest_date
)
UPDATE retail_sales_staging r
JOIN rfm_calc rfm
ON r.customer_id = rfm.customer_id
SET
    r.recency = rfm.recency_days,
    r.frequency = rfm.frequency,
    r.monetary = rfm.monetary;
```

Successfully created and updated the calculated RFM individual columns.

2. RFM Score Creation

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

SQL Query:

```sql
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,
```

```sql
-- 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 >= 1500 THEN 5
    WHEN monetary >= 1000 THEN 4
    WHEN monetary >= 500 THEN 3
    WHEN monetary >= 250 THEN 2
    ELSE 1
    END AS m_score
  FROM retail_sales_staging;
```

Saved each customer's RFM scores, combined them into a single RFM code, then use that code to assign a customer label.

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

UPDATE retail_sales_staging r
JOIN (
    SELECT customer_id,
      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,
      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 >= 1500 THEN 5
            WHEN monetary >= 1000 THEN 4
            WHEN monetary >= 500 THEN 3
            WHEN monetary >= 250 THEN 2
            ELSE 1
        END AS m_score
    FROM retail_sales_staging
    GROUP BY customer_id, recency, frequency, monetary
) scores
ON r.customer_id = scores.customer_id
SET
    r.r_score = scores.r_score,
    r.f_score = scores.f_score,
    r.m_score = scores.m_score;
```

Output:

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

3. RFM Code Creation

```
-- Merged the RFM scores into one code and added it as a column to help group and analyze customers better.
SELECT CONCAT(r_score, f_score, m_score) AS rfm_code
FROM retail_sales_staging;

ALTER TABLE retail_sales_staging
ADD column rfm_code int;

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

Note: At first, I stored the data type as int, I keep getting an error. After analyzing further, it should be a string (varchar or char) to use the rfm_code for labeling the segments.

Output:

rfm_code
411
213
111
212
211
211
211
211
412
311
211

4. Customer Segmentation

```sql
-- 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;
```

Output:

customer_id	segment_label
CUST001	Low Engagement
CUST002	At Risk
CUST003	At Risk
CUST004	At Risk
CUST005	At Risk
CUST006	At Risk
CUST007	At Risk
CUST008	At Risk
CUST009	Loyal Customers
CUST010	Low Engagement
CUST011	At Risk
CUST012	Low Engagement
CUST013	Big Spenders

```
-- 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);
```

Successfully created and updated the customer segmentation based on RFM codes.

We can now analyze customer behavior and purchasing patterns across segments.

a. **What percentage of customers belong to the high-value segment?**

SQL Query:

```
SELECT customer_segment,
ROUND(COUNT(customer_id) * 100 / (SELECT COUNT(customer_id) FROM retail_sales_staging),2) AS percent_of_customers
FROM retail_sales_staging
WHERE year(date) = 2023
GROUP BY customer_segment
ORDER BY percent_of_customers DESC;
```

Output:

customer_segment	percent_of_customers
At Risk	44.10
Low Engagement	37.70
Loyal Customers	9.10
Big Spenders	8.90

b. **Which RFM segments contribute the most to overall revenue?**

SQL Query:

```
SELECT DISTINCT(customer_segment), SUM(total_amount) AS total_sales
FROM retail_sales_staging
WHERE year(date) = 2023
GROUP BY customer_segment
ORDER BY total_sales DESC;
```

Output:

customer_segment	total_sales
Big Spenders	156000
At Risk	140105
Low Engagement	113675
Loyal Customers	44690

c. **What is the average RFM scores by segment?**

SQL Query:

```sql
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
WHERE year(date) = 2023
GROUP BY customer_segment;
```

Output:

customer_segment	avg_recency_score	avg_frequency_score	avg_monetary_score
Low Engagement	3.4354	1.0000	1.6702
At Risk	2.0000	1.0000	1.7370
Loyal Customers	5.0000	1.0000	2.1868
Big Spenders	2.7528	1.0000	5.0000

Created view to store selected metrics for further analysis.

SQL Query:

```sql
CREATE VIEW rfm_table AS
(
SELECT customer_id, customer_segment, rfm_code, recency, frequency, monetary
FROM retail_sales_staging
);
```

Output:

customer_id	customer_segment	rfm_code	recency	frequency	monetary
CUST001	Low Engagement	411	68	1	150
CUST002	At Risk	213	338	1	1000
CUST003	Low Engagement	111	383	1	30
CUST004	At Risk	212	255	1	500
CUST005	At Risk	211	270	1	100
CUST006	At Risk	211	281	1	30
CUST007	At Risk	211	324	1	50
CUST008	At Risk	211	343	1	100
CUST009	Low Engagement	412	49	1	600
CUST010	Low Engagement	311	116	1	200
CUST011	At Risk	211	351	1	100
CUST012	Low Engagement	411	93	1	75

SHARE

Data Visualization in Tableau

After cleaning and structuring the retail dataset using SQL, I imported the refined data into Tableau to develop interactive dashboards.

Purpose: These visualizations were designed to clearly present customer segmentation and sales performance insights, enabling stakeholders to explore trends and make data-driven decisions with ease.

Target Audience

This dashboard is designed for sales managers, marketing analysts, and e-commerce strategists seeking to enhance customer retention and maximize revenue.

Design and Interactivity

1. **Dashboard Dynamics**
- This dashboard provides users with the ability to navigate between the sales and customers dashboards easily.
- The charts and graphs are interactive, enabling users to filter data using charts.
- I refined the tooltips to make the key results easier to understand at a glance. The updated tooltips display only the most relevant details, helping users quickly interpret data without clutter or distraction.

2. **Data Filters**
- Allow users to filter data by appropriate information.
- These filters are designed to support flexible analysis, enabling stakeholders to drill down into specific customer behaviors or time periods for deeper insights.

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

1ˢᵗ Dashboard: KPI Overview

The dashboard highlights key metrics Total Sales, Total Quantity Sold, Total Transactions, and Average Order Value (AOV) using bold number displays (BANs) for quick reference. Each KPI is paired with a sparkline to show its performance trend over time.

Total Sales	Total Quantity Sold	Total Transaction	Average Order Value (AOV)
$454K	**2.51K**	**998**	**$455**
Jan — Dec	Jan — Dec	Jan — Dec	Jan — Dec
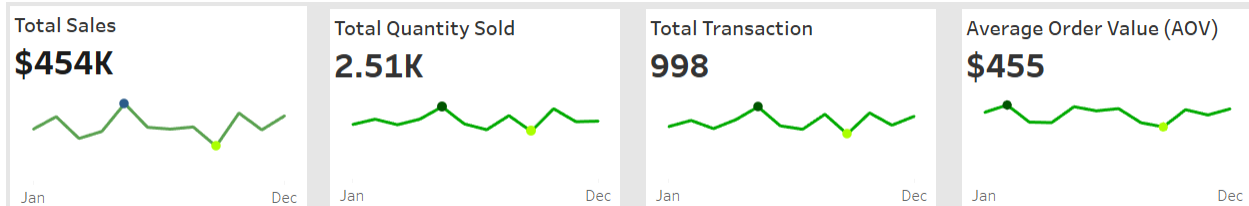
Insights:

Total Revenue:

- Total revenue is currently high, but the sparkline shows a recent decline. The current figure sits closer to the minimum ($24K) than the maximum ($53K) for the year, which may suggest a slowdown in customer activity and a need to boost engagement.

Total Quantity Sold:

- The total quantity sold has dropped from its peak during the mid-year period. Although sales are still active, the current volume is much closer to the yearly low, suggesting reduced customer demand or fewer large orders in recent months.

Total Transactions:

- Transaction volume peaked in May but has slowed down in September. It might be time to refresh promotions or explore why fewer orders are coming in.

Average Order Value(AOV)

- AOV reached its highest point in February at $518, and its lowest in September at $363. The current value is closer to the lower end, which could mean customers are making smaller purchases recently. This might be a good time to explore upselling strategies or limited-time bundles.

Sales Performance Analysis

a. **Monthly Sales Trend**

To display monthly sales and total amount, I used a **time series line chart** in Tableau based on transaction dates. I then changed the line style to "step" to highlight how sales values change from one month to the next. I also added an average reference line to show the overall sales benchmark, using single color with varying shades—darker for above average and lighterfor below average.

Monthly Sales Trend

● Above ● Below
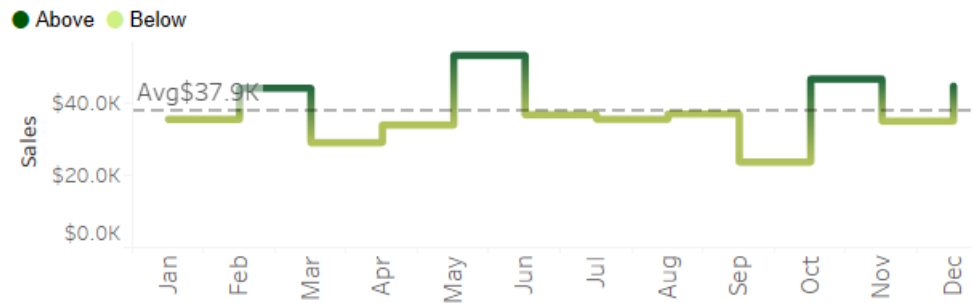
Insights:

- Sales peaked in May, experienced a dipped in September and sales bounced back in October.
- 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.

b. Transaction Frequency Over Months

I used a **bar chart** to show the number of transactions each month, making it easy to compare activity over time. To highlight the exact monthly totals, I added a Gantt bar overlay that displays the numeric values just above each bar. This makes it quick for users to see both the trend and the specific figures without needing to hover or check tooltips.

Transaction Frequency Over Months
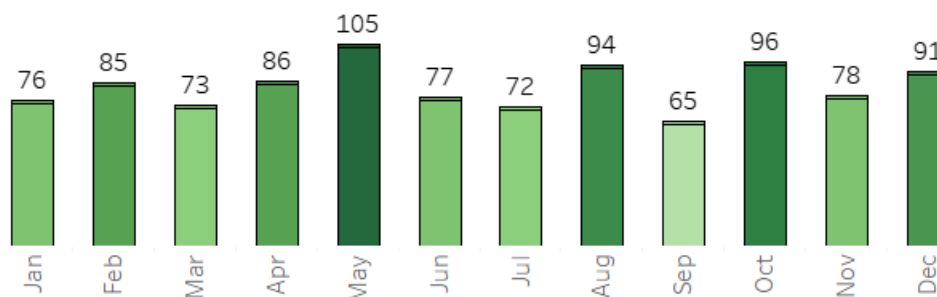
Insight:

- Transaction frequency reached its peak in May, indicating high customer activity, and experienced a noticeable dip in September.

c. Top-Performing Product Categories

I used **a horizontal bar chart** sorted from highest to lowest sales. To show the difference in performance, I applied a single color with varying shades—darker for higher sales and lighter for lower ones. This keeps the chart easy to read while still highlighting category rankings.

Product Category Distribution by Sales

Category	Sales
Electronics	$156.9K
Clothing	$155.6K
Beauty	$142.0K

Insight:

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

d. Gender Distribution by Product Category

I used a **horizontal stacked bar chart** to display the count of each gender within every product category. To enhance clarity, I applied a green-gold color palette—visually distinguishing product categories while keeping the chart easy to read.

Gender Distribution by Product Category

● Electronics ● Clothing ● Beauty

Gender	Electronics	Clothing	Beauty
Female	170	174	166
Male	171	177	140

Insight:

- Female customers show a stronger preference for clothing. While male customers also draw strong interest to electronics.

e. Product Categories Purchased by Age Group

I used a **stacked bar chart** to show which product categories are most popular across different age groups. To make the chart visually engaging and easy to interpret, I applied a green-gold color palette that clearly separates each category.

Product Categories Purchased by Age Group

● Electronics ● Clothing ● Beauty

Age Group	Electronics	Clothing	Beauty
Adult(31-45)	106	113	83
Middle Age(46-59)	104	109	95
Old(60+)	43	41	31
Teens(0-20)	23		24
Young(21-30)	65	72	73

Insight:

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

2nd Dashboard: Customer Segmentation (RFM)

a. Customer Segment Distribution

I used a heatmap to visualize customer segments, showing each segment's name, its percentage of the total, and the customer count. The color intensity helps highlight which segments are most notable, making it easy to spot key patterns at a glance.

Customer Segment Dustribution

At Risk	Loyal Customers
44.2%	9.1%
441	91

Low Engagement	Big Spenders
37.8%	8.9%
377	89

Insight:

- Only 18.0% of customers fall into high-value segments (Big Spenders and Loyal Customers), highlighting a small but important group that contributes significantly to sales. The majority of customers fall under the At Risk (44.2%) and Low Engagement (37.8%) segments, indicating a need for re-engagement strategies.

b. Average RFM Score Table

I created a table showing the average RFM scores for each segment, making it easy to spot which ones need immediate action.

Average RFM Score

| Customer Seg..	Avg. R Score	Avg. F Score	Avg. M Score
At Risk	2.00	1.00	1.74
Big Spenders	2.75	1.00	5.00
Low Engagement	3.45	1.00	1.67
Loyal Customers	5.00	1.00	2.19

Insight:
- Loyal Customers buy most recently, with the highest Recency score (5.0), while Big Spenders spend the most, with the top Monetary score (5.0). In contrast, Low Engagement and At-Risk segments have lower scores, showing they buy less often and spend less.

c. Customer Segment by Sales

I created a horizontal bar chart to show the segment which contributes most to sales, I sorted the segments from highest to lowest sales and used a green color scale—darker green shows higher sales.

Customer Segment by Sales

Segment	Sales
Big Spenders	$156.0K
At Risk	$140.1K
Low Engagement	$113.7K
Loyal Customers	$44.7K
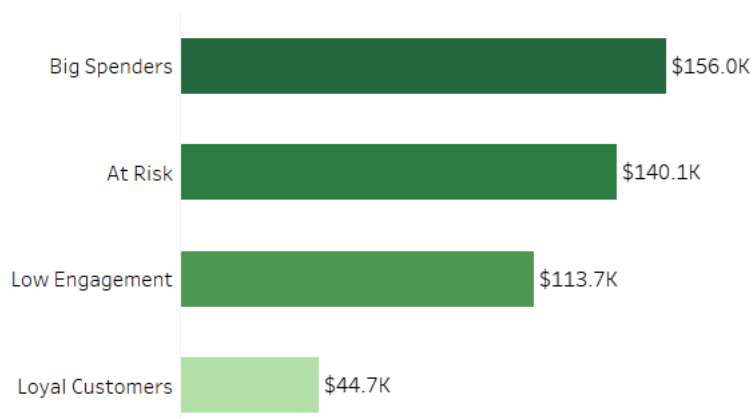
Insight:

- Analysis showed that the Big Spenders and At-Risk segments contributed the most to sales, while the Loyal Customers segment had a surprisingly minimal impact. This suggests that recent transactions may have come from less consistent buyers, highlighting an opportunity to convert these segments into more loyal ones through targeted re-engagement strategies.

d. Look Up Table

To easily Identify each customer's RFM segment, I created a lookup table, allowing for faster analysis and targeted strategy decisions based on behavior patterns

Search	(All)				▼

Customer Segmentation Table

Custom..	Customer ..	Rfm Co..	Recency	Frequency	Monetary
CUST024	Low Engage..	412	32	1	$300
CUST025	Loyal Custo..	511	5	1	$50
CUST026	Low Engage..	414	85	1	$1,000
CUST027	Low Engage..	311	150	1	$50
CUST028	At Risk	213	252	1	$500
CUST029	Low Engage..	311	135	1	$30
CUST030	Low Engage..	413	63	1	$900
CUST031	At Risk	214	222	1	$1,200
CUST032	At Risk	211	361	1	$90
CUST033	At Risk	211	283	1	$100
CUST034	Loyal Custo..	511	7	1	$150

ACT

Conclusion and Recommended Actions

Sales and Customer Demographics

Retail sales in 2023 showed strong performance overall. Sales and transaction volume peaked in May, dropped in September, and began to recover in October. May and December are strong sales months, likely due to holidays and seasonal events like school breaks. 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. Overall, electronics contributed the most to total sales, making it the top-performing product category for the year.

Recommended Action:

- Introduce targeted promotions for September and set clear performance goals.
- Re-engage inactive customers during low-activity months through strategic campaigns.
- Ensure balanced inventory and tailor promotions across product categories.

- Align marketing by audience: beauty for females, electronics for males, and unisex campaigns for clothing.
- Focus electronics promotions toward middle-aged and older customers; promote beauty products to younger buyers.

RFM Analysis

Only 18.0% of customers fall into high-value segments such as Big Spenders and Loyal Customers, while most customers fall into the At-Risk and Low Engagement groups. Though Big Spenders customers aren't purchase frequently, 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.

Recommended Actions:

- **High-Value Retention:** Strengthen relationships with Loyal Customers and Big Spenders by offering exclusive perks such as early access to products, premium support, or surprise rewards. This reinforces their value and encourages continued loyalty.
- **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:** Use feedback tools such as short surveys or exit polls to understand why customers stop purchasing and identify ways to enhance their journey.