### **RFM Analysis on E-Commerce Data**

[**E-Commerce Data**](https://www.kaggle.com/datasets/carrie1/ecommerce-data) dataset was used to perform RFM analysis with the following queries.

\*When calculating Recency, the latest order date was considered instead of the current date.

***RFM SCORE QUERY***

WITH recency AS

(

WITH max\_date AS (

SELECT

customerid,

MAX(invoicedate::date) AS max\_invoicedate

FROM

ecommerce\_data

WHERE

InvoiceNo NOT LIKE 'C%'

AND customerid IS NOT NULL

AND unitprice <> 0

GROUP BY

customerid

)

SELECT

customerid,

max\_invoicedate,

EXTRACT(DAY FROM AGE('2011-12-09'::date, max\_invoicedate)) AS recency

FROM

max\_date

),

frequency AS

(

SELECT

customerid,

COUNT(DISTINCT invoiceno) AS frequency

FROM

ecommerce\_data

WHERE

invoiceno NOT LIKE 'C%'

AND customerid IS NOT NULL

AND unitprice <> 0

GROUP BY

customerid

),

monetary AS

(

SELECT

customerid,

SUM(quantity \* unitprice) AS monetary

FROM

ecommerce\_data

WHERE

invoiceno NOT LIKE 'C%'

AND customerid IS NOT NULL

AND unitprice <> 0

GROUP BY

customerid

)

SELECT

r.customerid,

r.recency,

NTILE(5) OVER (ORDER BY recency DESC) AS recency\_score,

f.frequency,

-- Using CASE WHEN instead of NTILE for frequency

CASE

WHEN f.frequency = 1 THEN 1

WHEN f.frequency BETWEEN 2 AND 3 THEN 2

WHEN f.frequency BETWEEN 4 AND 5 THEN 3

WHEN f.frequency BETWEEN 6 AND 10 THEN 4

ELSE 5

END AS frequency\_score,

m.monetary,

NTILE(5) OVER (ORDER BY monetary DESC) AS monetary\_score,

-- Adjusted RFM Score calculation using the modified frequency score

CAST(

(NTILE(5) OVER (ORDER BY recency DESC) \* 100) +

(CASE

WHEN f.frequency = 1 THEN 1

WHEN f.frequency BETWEEN 2 AND 3 THEN 2

WHEN f.frequency BETWEEN 4 AND 5 THEN 3

WHEN f.frequency BETWEEN 6 AND 10 THEN 4

ELSE 5

END \* 10) +

(NTILE(5) OVER (ORDER BY monetary DESC))

AS VARCHAR

) AS rfmscore

FROM

recency AS r

INNER JOIN

frequency AS f ON r.customerid = f.customerid

INNER JOIN

monetary AS m ON m.customerid = r.customerid

ORDER BY

rfmscore DESC;

\*The code below has been written to create a table based on the RFMScore by assigning the most commonly used segments in the international literature."

WITH recency AS

(

WITH max\_date AS (

SELECT

customerid,

MAX(invoicedate::date) AS max\_invoicedate

FROM

ecommerce\_data

WHERE

InvoiceNo NOT LIKE 'C%' AND

customerid IS NOT NULL AND

unitprice <> 0

GROUP BY

customerid

)

SELECT

customerid,

max\_invoicedate,

EXTRACT(DAY FROM AGE('2011-12-09'::date, max\_invoicedate)) AS recency

FROM

max\_date

),

frequency AS

(

SELECT

customerid,

COUNT(DISTINCT invoiceno) AS frequency

FROM

ecommerce\_data

WHERE

invoiceno NOT LIKE 'C%'AND

customerid IS NOT NULL AND

unitprice <> 0

GROUP BY

customerid

),

monetary AS

(

SELECT

customerid,

SUM(quantity \* unitprice) AS monetary

FROM

ecommerce\_data

WHERE

invoiceno NOT LIKE 'C%'AND

customerid IS NOT NULL AND

unitprice <> 0

GROUP BY

customerid

),

RFMScores AS

(

SELECT

r.customerid,

r.recency,

NTILE(5) OVER (ORDER BY recency DESC) AS recency\_score,

f.frequency,

NTILE(5) OVER (ORDER BY frequency DESC) AS frequency\_score,

m.monetary,

NTILE(5) OVER (ORDER BY monetary DESC) AS monetary\_score,

CAST((NTILE(5) OVER (ORDER BY recency DESC) \* 100) + (NTILE(5) OVER (ORDER BY frequency DESC) \* 10) + (NTILE(5) OVER (ORDER BY monetary DESC)) AS VARCHAR) AS rfmscore

FROM

recency AS r

INNER JOIN

frequency AS f ON r.customerid = f.customerid

INNER JOIN

monetary AS m ON m.customerid = r.customerid

ORDER BY

rfmscore DESC

)

SELECT

customerid,

recency,

recency\_score,

frequency,

frequency\_score,

monetary,

monetary\_score,

CASE

WHEN rfmscore ~ '[1-2][1-2]' THEN 'Hibernating'

WHEN rfmscore ~ '[1-2][3-4]' THEN 'At Risk'

WHEN rfmscore ~ '[1-2]5' THEN 'Can''t Lose'

WHEN rfmscore ~ '3[1-2]' THEN 'About to Sleep'

WHEN rfmscore = '33' THEN 'Need Attention'

WHEN rfmscore ~ '[3-4][4-5]' THEN 'Loyal Customers'

WHEN rfmscore = '41' THEN 'Promising'

WHEN rfmscore = '51' THEN 'New Customers'

WHEN rfmscore ~ '[4-5][2-3]' THEN 'Potential Loyalists'

WHEN rfmscore ~ '5[4-5]' THEN 'Champions'

ELSE 'Undefined'

END AS segment

FROM

RFMScores

A graph with different colored bars

Description automatically generated

The RFM analysis of the e-commerce dataset reveals a distribution of customer segments, as shown in the table above. The largest group of customers falls under the "Hibernating" segment, indicating that these customers have not engaged with the company recently. This group represents a potential opportunity for re-engagement, as they may have previously shown interest but are now inactive. Tailored reactivation campaigns, such as special discounts or targeted reminders, could help bring these customers back into the active customer base.

The second largest segment is "At Risk," composed of customers who have been active in the past but may now be close to disengaging. This segment is critical to address as it represents customers who might still have interest but require prompt action to maintain their loyalty. To retain this group, personalized offers, re-engagement emails, or exclusive benefits could be deployed to rekindle their relationship with the brand.

On a more positive note, the "Loyal Customers" segment showcases a core group of consistently engaged customers who contribute significantly to the company’s revenue. This segment represents the company’s foundational customer base and serves as a strong indicator of customer potential. Efforts to further nurture and expand this loyal group could involve loyalty programs, exclusive rewards, or enhanced customer experiences. By increasing the number of customers in this segment, the company could boost customer lifetime value and foster long-term growth.

The analysis highlights key areas for strategic focus: reactivating inactive customers, retaining at-risk customers, and growing the loyal customer base. By addressing the needs and preferences of each segment, the company can build a more engaged, profitable, and sustainable customer portfolio.