IE6400 Foundations of Data Analytics Engineering

Project 1

CUSTOMER SEGMENTATION USING RFM ANALYSIS

Report

STUDENT NAME

Varaalakshime Vigneswara Pandiajan

Anita Janie Christdoss Chelladurai

Anoushka Karan Pakhare

INTRODUCTION

In today's fast-moving and highly competitive world of online shopping, understanding customers and their behavior has become one of the most valuable advantages a business can possess. With countless choices available to consumers and increasing pressure on companies to remain relevant, the ability to analyze how customers interact with a digital store is essential. The eCommerce dataset used in this project provides a rich opportunity to explore these behaviors through the RFM method an analytical framework that examines three core factors: how recently a customer made a purchase (Recency), how often they buy (Frequency), and how much money they spend overall (Monetary). By studying these three dimensions, businesses can group customers based on their purchasing habits and design smarter, more personalized marketing strategies.

As online markets continue to expand, businesses can no longer rely on broad, one-size-fits-all marketing approaches. Personalization is now a key driver of customer satisfaction and retention. RFM analysis supports this need by offering a structured way to categorize customers into meaningful groups, each representing different behaviors and levels of engagement. By breaking down the dataset using these metrics, the analysis uncovers hidden trends in customer activity, such as who shops frequently, who has high spending power, and who may be losing interest. These patterns provide critical insights into how customers interact with the store and highlight which individuals or groups contribute most to a company's revenue.

The main goal of this project is to use RFM analysis to create actionable and meaningful customer segments that reflect real shopping behavior. Each segment is built from customers who share similar purchasing characteristics, allowing businesses to deliver tailored experiences that match the needs and expectations of different groups. This segmentation not only helps companies develop better promotional strategies and personalized offers but also strengthens customer retention by targeting the right people with the right message at the right time. Moreover, identifying customers with high monetary values enables businesses to recognize their most valuable segments and allocate resources more effectively.

When the three RFM components are combined, they form a comprehensive score that summarizes a customer's overall value and engagement level. These scores help reveal the full distribution of customer behavior across the entire dataset and serve as a foundation for advanced analysis techniques such as clustering. By leveraging RFM scores, the project builds clear customer groups that mirror patterns in recency, frequency, and spending, ultimately supporting stronger decision-making.

In summary, this project uses RFM analysis as a powerful tool to better understand the customers in an eCommerce environment. The insights gained from this segmentation help businesses personalize their marketing strategies, improve customer satisfaction, optimize engagement, and reinforce their competitive position in the growing online marketplace. By transforming raw transactional data into meaningful customer intelligence, RFM analysis becomes a vital asset in shaping the future of digital commerce.

Implementation

1) Data Preprocessing

The dataset underwent essential preprocessing to ensure data quality and analytical readiness. The invoice dataset contains 8 columns including InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, and Country.

❖ Missing Value Treatment

Initial assessment revealed 1,454 missing values in Description and 135,080 missing values in CustomerID. The following steps were applied:

- Removed rows where all values were missing using dropna(how='all')
- Imputed numeric columns with median values to maintain distribution
- Filled categorical columns with mode values to preserve dominant patterns
- Successfully reduced missing values to zero across all columns

```
[4]: #Checking missing values
     df.isnull().sum()
```

```
[4]: InvoiceNo          0
     StockCode          0
     Description     1454
     Quantity           0
     InvoiceDate        0
     UnitPrice          0
     CustomerID    135080
     Country            0
     dtype: int64
```

Fig 1.Check Missing Values

```
[6]: # Handling missing values

     # Drop rows where all values are missing
     df.dropna(how='all', inplace=True)

     # Fill numeric columns with their median
     num_cols = df.select_dtypes(include=[np.number]).columns
     df[num_cols] = df[num_cols].apply(lambda x: x.fillna(x.median()))

     # Fill categorical columns with their mode
     cat_cols = df.select_dtypes(exclude=[np.number]).columns
     for col in cat_cols:
         mode_value = df[col].mode(dropna=True)
         if not mode_value.empty:
             df[col] = df[col].fillna(mode_value[0])
```

```
[7]: df.isnull().sum()
```

```
[7]: InvoiceNo      0
     StockCode      0
     Description    0
     Quantity       0
     InvoiceDate    0
     UnitPrice      0
     CustomerID     0
     Country        0
     dtype: int64
```

Fig 2.Handling Missing Values

❖ Data Type Conversion

Date columns were automatically detected and converted to proper datetime format. The InvoiceDate column was identified and converted using pd.to_datetime(), and a new order_date column was created for date-based analysis.

```
[8]: #  Data type conversions
     if 'order_date' in df.columns:
         df['order_date'] = pd.to_datetime(df['order_date'], errors='coerce')

     df.dtypes
```

```
[8]: InvoiceNo       object
     StockCode       object
     Description     object
     Quantity         int64
     InvoiceDate     object
     UnitPrice      float64
     CustomerID     float64
     Country         object
     dtype: object
```

Fig 3.Data type Conversion

❖ Dataset Coverage

The preprocessed dataset spans from December 1, 2010 to December 9, 2011, covering approximately one year of transaction data. Final data types include object types for categorical fields and numeric types (int64, float64) for quantitative measures, ensuring optimal performance for analysis.

```
#Detect correct date column and find dataset period

import pandas as pd

possible_date_cols = [col for col in df.columns if 'date' in col.lower()]

if not possible_date_cols:
    possible_date_cols = [col for col in df.columns if 'invoice' in col.lower() and df[col].astype(str).str.contains(r'\d{4}-\d{2}-\d{2}|/').any()]

if possible_date_cols:
    date_col = possible_date_cols[0]
    print(f"Detected date column: {date_col}")
    df[date_col] = pd.to_datetime(df[date_col], errors='coerce')
    df = df.dropna(subset=[date_col])
    df['order_date'] = df[date_col].dt.date
    print(f"Dataset covers from {df['order_date'].min()} to {df['order_date'].max()}")
else:
    print("No valid date column detected. Please check your column names manually using df.columns")
```

```
Detected date column: InvoiceDate
Dataset covers from 2010-12-01 to 2011-12-09
```

Fig 4.Find Data Period 1

A data overview helps us understand the bigger picture of the dataset before diving into analysis. It involves looking at the structure of the data, checking its quality, exploring each feature, and making sure we know what information is available. To begin, we answer a few key questions:

❖ What is the size of the dataset in terms of the number of rows and columns?

The original dataset contains 541,909 rows and 8 columns.
After cleaning and removing invalid entries and fixing missing values—the dataset becomes 539,392 rows and 8 columns.
Knowing the size helps us understand the scale of the data and how much of it needed correction during preprocessing.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

Fig 5.Print head of the data

❖ Can you provide a brief description of each column in the dataset?

The dataset is made up of eight main columns, each providing important information about the transactions:

1. InvoiceNo: A unique ID for each order. Orders with a “C” indicate cancellations.
2. StockCode: A unique alphanumeric code for each product.
3. Description: A text description of the product.
4. Quantity: The number of units of a product purchased in a single order.
5. InvoiceDate: The date and time when the order was placed.
6. UnitPrice: The price of one unit of the product.
7. CustomerID: A unique identifier for each customer, used to track their purchases.
8. Country: The country from which the order was made.

The dataset includes both numerical columns (like Quantity, UnitPrice, and CustomerID) and categorical/text columns (such as Description, Country, and StockCode). Understanding these columns helps us interpret the dataset correctly and prepares us for further analysis.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  object
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
```

Fig 6.Data Info

❖ What is the period covered by this dataset?

The dataset covers a period from 2010-12-01 08:26:00 to 2011-12-09 12:50:00. There are 4372 unique customers in the dataset. It is crucial to consider such an understanding of the timespan within which the data was collected when it comes to understanding changes in customer behavior during those periods.

```
Detected date column: InvoiceDate
Dataset covers from 2010-12-01 to 2011-12-09
```

```
The dataset has 541909 rows and 13 columns after data cleaning
            Quantity                    InvoiceDate      UnitPrice  \
count  541909.000000                         541909  541909.000000
mean        9.552250  2011-07-04 13:34:57.156386048       4.611114
min    -80995.000000            2010-12-01 08:26:00  -11062.060000
25%         1.000000            2011-03-28 11:34:00       1.250000
50%         3.000000            2011-07-19 17:17:00       2.080000
75%        10.000000            2011-10-19 11:27:00       4.130000
max     80995.000000            2011-12-09 12:50:00   38970.000000
std       218.081158                            NaN      96.759853

          CustomerID     order_hour        Revenue
count  541909.000000  541909.000000  541909.000000
mean    15253.867397      13.078729      17.987795
min     12346.000000       6.000000 -168469.600000
25%     14367.000000      11.000000       3.400000
50%     15152.000000      13.000000       9.750000
75%     16255.000000      15.000000      17.400000
max     18287.000000      20.000000  168469.600000
std      1485.905852       2.443270     378.810824
```

Fig 7.Find Data Period

2) RFM Calculation

We calculated RFM metrics to understand customer purchasing behavior. First, we identified or created a monetary value for each transaction (using an existing total column, or by computing *quantity* × *unit price*). Then, for each customer, we calculated:

- **Recency:** how many days since their latest purchase
- **Frequency:** how many times they purchased
- **Monetary:** how much they spent in total

RFM (first 10 customers):

	CustomerID	recency_days	frequency	monetary	first_purchase	last_purchase	avg_order_value	active_days
0	12346.0	326	2	2.08	2011-01-18 10:01:00	2011-01-18 10:17:00	1.040000	0
1	12347.0	2	182	481.21	2010-12-07 14:57:00	2011-12-07 15:52:00	2.644011	365
2	12348.0	75	31	178.71	2010-12-16 19:09:00	2011-09-25 13:13:00	5.764839	282
3	12349.0	19	73	605.10	2011-11-21 09:51:00	2011-11-21 09:51:00	8.289041	0
4	12350.0	310	17	65.30	2011-02-02 16:01:00	2011-02-02 16:01:00	3.841176	0
5	12352.0	36	95	2211.10	2011-02-16 12:33:00	2011-11-03 14:37:00	23.274737	260
6	12353.0	204	4	24.30	2011-05-19 17:47:00	2011-05-19 17:47:00	6.075000	0
7	12354.0	232	58	261.22	2011-04-21 13:11:00	2011-04-21 13:11:00	4.503793	0
8	12355.0	214	13	54.65	2011-05-09 13:49:00	2011-05-09 13:49:00	4.203846	0
9	12356.0	23	59	188.87	2011-01-18 09:50:00	2011-11-17 08:40:00	3.201186	302

RFM scores added to `customers` summary.

Fig 8.RFM CALCULATION

❖ Customer Analysis

In this customer analysis, we focus on understanding customers, with an important measure being customer churn, also called customer attrition, which is the loss of customers for any reason. The calculated churn rate is 34.42%. Generally, a churn rate above 15% is considered high, but this depends on the industry and business context. Some industries naturally have higher churn due to their market or business model. It is also important to look at trends over time, find out why customers leave, and compare the churn rate with competitors or industry standards. In this section, we will answer some basic questions about customer analysis.

1. How many unique customers are there in the dataset?

   There are 4372 unique customers in the dataset

```
1) Unique customers: 4372
```

Fig 9.Customer Analysis 1

2. What is the distribution of the number of orders per customer?

   To understand the distribution of the number of orders per customer, we can infer the following:

   Mean (Average Number of Orders per Customer): The mean value of 123.94 represents the average number of orders per customer.

   Standard Deviation (Variability): The standard deviation (2058.88) measures the amount of variation or dispersion in the number of orders per customer.

   Percentiles (25%, 50%, 75%): These values provide insights into the distribution of orders.
   For example, 25% of customers have made 17 order or fewer (25th percentile), 50% of customers have made 42 orders or fewer (median), and 75% of customers have made 102 orders or fewer (75th percentile).

   Understanding the distribution of orders per customer is valuable for businesses in tailoring marketing, especially considering the variations in customer behavior.

```
2) Orders per customer — summary stats:
count      4372.000000
mean        123.949909
std        2058.880484
min           1.000000
25%          17.000000
50%          42.000000
75%         102.000000
max      135358.000000
Name: frequency, dtype: float64

frequency
1     79
2     59
3     53
4     55
5     61
6     78
7     72
8     67
9     67
10    74
11    70
12    72
13    57
14    62
15    61
16    67
17    56
18    55
19    62
20    49
Name: count, dtype: int64
```

Fig 10.Customer Analysis 2

3. Can you identify the top 5 customers who have made the most purchases by order count?

   To identify most frequent buyers and making the platform more engaging for them by giving them special treatment and/or facilities like premium membership, free delivery, x% off on products, early access, etc., is a common practice to hold on to a customer by businesses. So, to identify the top 5 customers we sorted the data in descending order by "Orders" and displayed the top 5 rows.

3) Top 5 customers by order count:

	CustomerID	frequency	monetary	avg_order_value
2073	15152.0	135358	1091966.33	8.067246
4042	17841.0	7983	20333.18	2.547060
1895	14911.0	5903	31060.66	5.261843
1300	14096.0	5128	41376.33	8.068707
330	12748.0	4642	15115.60	3.256269

Fig 11.Customer Analysis 3

❖ Product Analysis

Product analysis is essential for businesses as it provides insights into how products perform, support growth strategies, influence inventory decisions, respond to market trends, and guide business choices. A key objective in business is identifying top-selling products.

1. What are the top 10 most frequently purchased products?

   To determine which products perform best, we examined the 10 most frequently ordered items and highlighted those with the highest number of orders alongside their descriptions.

```
Top 10 most frequently purchased products (by transaction lines):

Description
WHITE HANGING HEART T-LIGHT HOLDER    3823
REGENCY CAKESTAND 3 TIER              2200
JUMBO BAG RED RETROSPOT               2159
PARTY BUNTING                         1727
LUNCH BAG RED RETROSPOT               1638
ASSORTED COLOUR BIRD ORNAMENT         1501
SET OF 3 CAKE TINS PANTRY DESIGN      1473
PACK OF 72 RETROSPOT CAKE CASES       1385
LUNCH BAG  BLACK SKULL.               1350
NATURAL SLATE HEART CHALKBOARD        1280
Name: count, dtype: int64
```

Fig 12.Product Analysis 1

2. What is the average price of products in the dataset?

   Calculating the average price of a product is crucial for understanding its typical market value.

```
Average product price: 4.61

Description
DOTCOM POSTAGE    206245.48
Name: Revenue, dtype: float64


Average unit price (column 'UnitPrice'): 4.61

No product category column detected. Skipping category revenue analysis.
```

Fig 13.Product Analysis 2

3. Can you find out which product category generates the highest revenue?

   Revenue is another critical business metric. To assess revenue, we created an additional column that calculates total revenue by multiplying product quantity by unit price. To identify the top 10 revenue-generating products, we grouped items by description, calculated their total revenue, sorted the results in descending order, and selected the first 10 entries.

   In conclusion, product analysis is a comprehensive approach that supports strategic decision-making, resource allocation, and overall business performance. These analytical techniques not only enhance business operations but also identify product weaknesses and deepen customer understanding.

❖ Timeframe Analysis

1. Is there a specific day of the week or time of day when most orders are placed?

To understand when customers are most active, we conducted a time analysis using the order date field. We first converted the order timestamp into usable components such as weekday and hour. By analyzing order counts by weekday, we identified which days experience the highest and lowest purchasing activity. We then examined order distribution across different hours of the day to uncover peak transaction times. Additionally, we checked for shipping or completion timestamps to estimate average processing time when available. Finally, we generated monthly order counts to observe seasonal trends and identify periods of high or low demand.

```
1) Orders by day of week (counts):

order_dayofweek
Thursday     103857
Tuesday      101808
Monday        95111
Wednesday     94565
Friday        82193
Sunday        64375
Name: count, dtype: int64
```

Fig 14.TimeFrame Analysis 1

```
2) Orders by hour of day (counts):

order_hour
6        41
7       383
8      8909
9     34332
10    49037
11    57674
12    78709
13    72259
14    67471
15    77519
16    54516
17    28509
18     7974
19     3705
20      871
Name: count, dtype: int64
```

Fig 15.TimeFrame Analysis 2

By organizing invoices into two-hour intervals, we determined that most orders occur between 10:00 AM and 4:00 PM, with peak ordering specifically between 12:00 PM and 2:00 PM

2. What is the average order processing time?

Average order processing time cannot be determined from available data. To calculate processing duration, we would require shipping date and time information following order placement. The difference between order placement and shipment would provide processing time.

3. Are there any seasonal trends in the dataset?

To identify seasonal patterns, we grouped data by month to determine monthly order volumes. The analysis reveals seasonal trends with orders increasing toward year-end, with November recording the highest order count (84711 orders). This likely reflects holiday gift purchasing and seasonal promotions such as post-Thanksgiving Black Friday sales.

```
order_month
2010-12    42481
2011-01    35147
2011-02    27707
2011-03    36748
2011-04    29916
2011-05    37030
2011-06    36874
2011-07    39518
2011-08    35284
2011-09    50226
2011-10    60742
2011-11    84711
2011-12    25525
Freq: M, Name: count, dtype: int64
```

Fig 16.TimeFrame Analysis 3

❖ **Geographical Analysis**

1. Can you determine the top 5 countries with the highest number of orders?

Geographic analysis utilizes country data from invoices to determine order origins, helping businesses identify potential future markets. This enables more precise targeted advertising and sustained customer engagement.

The five countries with highest order volumes are: United Kingdom (495478 orders), Germany (9495 orders), France (8557 orders), EIRE/Ireland (8196 orders), and Spain (2533 orders).

Excluding the United Kingdom, the next five countries show more comparable order volumes, with the Netherlands ranking sixth.

```
Top 5 countries by number of orders:

Country
United Kingdom    495478
Germany             9495
France              8557
EIRE                8196
Spain               2533
Name: count, dtype: int64
```

Fig 17.Geographic Analysis 1

2. Is there a correlation between the customer's country and the average order value?

The avg_order_value varies across countries:

Highest: Portugal – 8.58 per order

Lowest: Netherlands – 2.73 per order

This shows that customer country does influence average order value, but it is not strictly proportional to the number of orders. For example, the UK has the highest number of orders but a moderate average order value of 4.53.

Countries with fewer orders (like Portugal and EIRE) can have higher average order values, indicating smaller markets but higher per-order spending.

Orders and average order value by country (top 10):

Country	num_orders	avg_order_value	total_revenue
United Kingdom	495478	4.532422	2245715.474
Germany	9495	3.966930	37666.000
France	8557	5.028864	43031.990
EIRE	8196	5.911077	48447.190
Spain	2533	4.987544	12633.450
Netherlands	2371	2.738317	6492.550
Belgium	2069	3.644335	7540.130
Switzerland	2002	3.403442	6813.690
Portugal	1519	8.582976	13037.540
Australia	1259	3.220612	4054.750

Correlation is categorical; inspect 'avg_order_value' column above to compare countries.

Fig 18.Geographic Analysis 2

3) RFM Segmentation

❖ simple rule based segments

Customer segmentation was performed using RFM (Recency, Frequency, Monetary) analysis, a proven method for identifying customer value based on their purchasing behavior. Three key metrics were calculated for each customer:

- Recency (R): Days since the last purchase
- Frequency (F): Total number of transactions
- Monetary (M): Total spending amount

Each metric was scored on a scale of 1-5, with higher scores indicating more valuable customer behavior. These scores were then combined to create comprehensive RFM scores for segmentation.

Segmentation Rules

Customers were classified into five distinct segments based on their RFM scores using rule-based thresholds:

- Champions: High performers across all metrics ($R \geq 4$, $F \geq 4$, $M \geq 4$) - Most valuable customers with recent, frequent, and high-value purchases
- Loyal: Consistent customers with strong frequency and monetary scores ($R \geq 3$, $F \geq 3$, $M \geq 3$) - Reliable customers who purchase regularly
- At Risk: Low recency but high frequency ($R \leq 2$, $F \geq 4$) - Previously active customers who haven't purchased recently
- Lost: Low scores across all metrics ($R \leq 2$, $F \leq 2$, $M \leq 2$) - Inactive customers requiring re-engagement strategies
- Others: Customers not falling into the above categories - Requires further analysis

Segment Distribution

The analysis revealed the following customer distribution:

- Others: 1,354 customers (31.3%)
- Champions: 998 customers (23.1%)
- Lost: 909 customers (21.0%)
- Loyal: 806 customers (18.6%)
- At Risk: 305 customers (7.1%)

```
Segment counts:
segment
Others       1354
Champions     998
Lost          909
Loyal         806
At Risk       305
Name: count, dtype: int64
```

Sample customers per segment:

CustomerID	recency_days	frequency	monetary	first_purchase	last_purchase	avg_order_value	active_days	r_score	f_score	m_score	rfm_score	rfm_sum	segment
12346.0	326	2	2.08	2011-01-18 10:01:00	2011-01-18 10:17:00	1.040000	0	1	1	1	111	3	Lost
12347.0	2	182	481.21	2010-12-07 14:57:00	2011-12-07 15:52:00	2.644011	365	5	5	5	555	15	Champions
12348.0	75	31	178.71	2010-12-16 19:09:00	2011-09-25 13:13:00	5.764839	282	2	3	3	233	8	Others
12349.0	19	73	605.10	2011-11-21 09:51:00	2011-11-21 09:51:00	8.289041	0	4	4	5	445	13	Champions
12350.0	310	17	65.30	2011-02-02 16:01:00	2011-02-02 16:01:00	3.841176	0	1	2	2	122	5	Lost
12352.0	36	95	2211.10	2011-02-16 12:33:00	2011-11-03 14:37:00	23.274737	260	3	4	5	345	12	Loyal
12353.0	204	4	24.30	2011-05-19 17:47:00	2011-05-19 17:47:00	6.075000	0	1	1	1	111	3	Lost
12354.0	232	58	261.22	2011-04-21 13:11:00	2011-04-21 13:11:00	4.503793	0	1	3	4	134	8	Others
12356.0	23	59	188.87	2011-01-18 09:50:00	2011-11-17 08:40:00	3.201186	302	4	4	4	444	12	Champions
12357.0	33	131	438.67	2011-11-06 16:07:00	2011-11-06 16:07:00	3.348626	0	3	5	5	355	13	Loyal
12358.0	2	19	157.21	2011-07-12 10:04:00	2011-12-08 10:26:00	8.274211	149	5	2	3	523	10	Others
12360.0	52	129	457.91	2011-05-23 09:43:00	2011-10-18 15:22:00	3.549690	148	3	5	5	355	13	Loyal

Fig 19.Simple Rule Based Segments

❖ **RFM scores to each customer based on their quartiles**

The RFM analysis began with temporal data conversion, transforming InvoiceDate to datetime format. A snapshot date was established as one day after the last transaction to accurately measure customer recency. Transaction data was then aggregated by CustomerID to compute three fundamental metrics: recency_days (days since last purchase), frequency (total number of transactions), and monetary (total spending).

Scoring Method

Each RFM metric was scored using pandas' qcut function with 4 quartiles (labels 1-4):

Recency Score: Days since last purchase were divided into quartiles with inverted scoring—lower recency values (recent customers) received higher scores (4,3,2,1), ensuring recent purchasers ranked higher.

Frequency Score: Transaction counts were ranked using the 'first' method and segmented into quartiles (1,2,3,4), with frequent buyers earning higher scores.

Monetary Score: Total spending was divided into quartiles (1,2,3,4), rewarding high-value customers with elevated scores.

[31]:
```
rfm['r_score'] = pd.qcut(rfm['recency_days'], 4, labels=[4,3,2,1]).astype(int)  # lower recency = higher score
rfm['f_score'] = pd.qcut(rfm['frequency'].rank(method='first'), 4, labels=[1,2,3,4]).astype(int)
rfm['m_score'] = pd.qcut(rfm['monetary'], 4, labels=[1,2,3,4]).astype(int)

rfm[['CustomerID', 'recency_days', 'frequency', 'monetary', 'r_score', 'f_score', 'm_score']].head()
```

[31]:

	CustomerID	recency_days	frequency	monetary	r_score	f_score	m_score
0	12346.0	326	0	0.00	1	1	1
1	12347.0	2	0	4310.00	4	1	4
2	12348.0	75	0	1797.24	2	1	4
3	12349.0	19	0	1757.55	3	1	4
4	12350.0	310	0	334.40	1	1	2

Fig 20.Scoring Method

Composite Metrics

Two composite metrics were created for comprehensive customer evaluation:

RFM Score String: Individual R, F, and M scores were concatenated as strings (e.g., "414" = r_score:4, f_score:1, m_score:4), enabling granular pattern recognition for segmentation rules.

RFM Sum: The three scores were summed to create rfm_sum (range: 3-12), providing a simplified overall customer value indicator for quick assessment and ranking.

```
[32]: rfm['rfm_score'] = rfm['r_score'].astype(str) + rfm['f_score'].astype(str) + rfm['m_score'].astype(str)
      rfm['rfm_sum'] = rfm['r_score'] + rfm['f_score'] + rfm['m_score']

      rfm[['CustomerID', 'r_score', 'f_score', 'm_score', 'rfm_score', 'rfm_sum']].head()
```

[32]:

	CustomerID	r_score	f_score	m_score	rfm_score	rfm_sum
0	12346.0	1	1	1	111	3
1	12347.0	4	1	4	414	9
2	12348.0	2	1	4	214	7
3	12349.0	3	1	4	314	8
4	12350.0	1	1	2	112	4

Fig 21.Composite Metrics

4) Customer Segmentation

The clustering analysis utilized the three RFM scores (r_score, f_score, m_score) as input features. These features were standardized using StandardScaler to ensure equal weighting across all dimensions, transforming the dataset of 4,372 customers into normalized values with zero mean and unit variance.

Optimal Cluster Determination

Two validation methods were employed to identify the optimal number of clusters:

Elbow Method: Computed Sum of Squared Errors (SSE) for cluster ranges from 2 to 10. The SSE curve showed a steep decline from 2 to 5 clusters, with diminishing returns beyond k=5, suggesting an "elbow" around 4-5 clusters.

Silhouette Score: Evaluated cluster quality by measuring how similar objects are to their own cluster compared to other clusters. The analysis revealed peak silhouette scores at k=4 (0.385) and k=9 (0.387), with k=4 showing strong separation while maintaining interpretability.

```
[35]: from sklearn.cluster import KMeans
      from sklearn.metrics import silhouette_score
      import matplotlib.pyplot as plt

      # Elbow method
      sse = []
      sil_scores = []
      K_range = range(2, 11)

      for k in K_range:
          kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
          kmeans.fit(rfm_scaled)
          sse.append(kmeans.inertia_)
          sil_scores.append(silhouette_score(rfm_scaled, kmeans.labels_))

      # Plot Elbow method and Silhouette scores
      plt.figure(figsize=(12,5))

      plt.subplot(1,2,1)
      plt.plot(K_range, sse, marker='o')
      plt.xlabel('Number of clusters')
      plt.ylabel('SSE')
      plt.title('Elbow Method')

      plt.subplot(1,2,2)
      plt.plot(K_range, sil_scores, marker='o')
      plt.xlabel('Number of clusters')
      plt.ylabel('Silhouette Score')
      plt.title('Silhouette Score')

      plt.show()
```

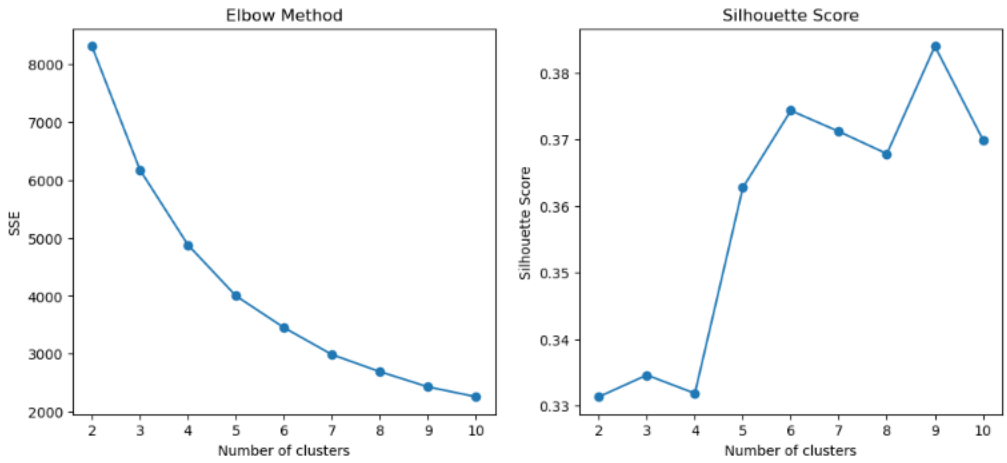

Fig 22.Optimal Cluster Determination

Final Clustering Implementation

Based on the combined analysis, K-Means was implemented with k=4 clusters using random_state=42 for reproducibility. The resulting cluster distribution showed:

- Cluster 1: 1,590 customers (36.4%)
- Cluster 2: 1,122 customers (25.7%)
- Cluster 3: 991 customers (22.7%)
- Cluster 0: 669 customers (15.3%)

```
[36]: # Fit K-Means with optimal clusters
k = 4
kmeans = KMeans(n_clusters=k, random_state=42)
rfm['cluster'] = kmeans.fit_predict(rfm_scaled)

print(rfm['cluster'].value_counts())
```

```
cluster
1    1590
2    1122
3     991
0     669
Name: count, dtype: int64
```

Fig 23.Cluster Implementation

5) Segment Profiling

Cluster Profiling

Each cluster was profiled by computing mean values for all RFM metrics and scores:

Cluster 1: Highest recency (215 days), minimal frequency (0.5), and low monetary value (315.8) - represents dormant or lost customers with rfm_sum of 6.9.

Cluster 3: Moderate recency (161 days), very low frequency (0.1), but highest monetary value (516.2) - indicates high-value infrequent purchasers with rfm_sum of 6.9.

Cluster 0: Low recency (29.7 days), minimal frequency (0.5), and very high monetary spending (5194.3) - represents recent high-value customers with rfm_sum of 9.9.

Cluster 2: Moderate recency (167 days), minimal frequency (0.0), and lowest monetary value (682.7) - low engagement customers with rfm_sum of 5.1.

The clusters were sorted by rfm_sum_mean in descending order to prioritize high-value segments, enabling targeted marketing strategies based on distinct customer behavioral patterns.

[38]:
```
# Group by cluster
cluster_summary = rfm.groupby('cluster').agg({
    'recency_days': ['mean', 'min', 'max'],
    'frequency': ['mean', 'min', 'max'],
    'monetary': ['mean', 'min', 'max'],
    'r_score': 'mean',
    'f_score': 'mean',
    'm_score': 'mean',
    'rfm_sum': ['mean', 'min', 'max'],
    'CustomerID': 'count'  # Number of customers in each cluster
})

# Clean column names
cluster_summary.columns = ['_'.join(col).strip() for col in cluster_summary.columns.values]

# Sort clusters by rfm_sum_mean descending
cluster_summary = cluster_summary.sort_values('rfm_sum_mean', ascending=False)

cluster_summary
```

[38]:

cluster	recency_days_mean	recency_days_min	recency_days_max	frequency_mean	frequency_min	frequency_max	monetary_mean	monetary_min	monetary_max
1	21.547799	1	140	0.489308	0	167	5194.335082	296.71	1452366.36
3	161.625631	18	374	0.169526	0	3	516.167297	-4287.63	21535.90
0	29.757848	1	143	0.023916	0	2	315.779656	-134.80	646.92
2	167.630125	18	374	0.000000	0	0	682.688745	-1592.49	11056.93

Fig 24.Cluster Profiling

Business-Oriented Segment Labeling

To enhance interpretability for business stakeholders, clusters were assigned descriptive labels based on their behavioral characteristics using a custom function:

Champions (Cluster 0): Customers with high recency scores (r_score ≥ 3), high frequency (f_score ≥ 3), and high monetary values (m_score ≥ 3) - the most valuable segment requiring premium retention strategies.

Potential Loyalists (Cluster 1 & 3): Customers showing moderate engagement with r_score ≥ 3 and f_score < 2 - promising segment with growth potential through targeted nurturing campaigns.

Frequent but Recent Drop-off: Customers with low recency (r_score < 2) but high frequency (f_score ≥ 3) - at-risk valuable customers requiring immediate re-engagement efforts.

At Risk / Low Value: All other customers not meeting the above criteria - lowest priority segment requiring cost-effective mass marketing approaches.

These business-oriented labels facilitate actionable segmentation strategies aligned with customer lifetime value and engagement levels.

[39]:
```
def label_cluster(row):
    if row['r_score_mean'] >= 3 and row['f_score_mean'] >= 3 and row['m_score_mean'] >= 3:
        return 'Champions'
    elif row['r_score_mean'] >= 3 and row['f_score_mean'] <= 2:
        return 'Potential Loyalists'
    elif row['r_score_mean'] <= 2 and row['f_score_mean'] >= 3:
        return 'Frequent but Recent Drop-off'
    else:
        return 'At Risk / Low Value'

# Apply labels to clusters
cluster_summary['segment_label'] = cluster_summary.apply(label_cluster, axis=1)

cluster_summary
```

[39]:

ary_mean	monetary_min	monetary_max	r_score_mean	f_score_mean	m_score_mean	rfm_sum_mean	rfm_sum_min	rfm_sum_max	CustomerID_count	segment_label
94.335082	296.71	1452366.36	3.480503	2.830189	3.602516	9.913208	8	12	1590	At Risk / Low Value
16.167297	-4287.63	21535.90	1.575177	3.458123	1.856710	6.890010	5	9	991	Frequent but Recent Drop-off
15.779656	-134.80	646.92	3.240658	2.130045	1.487294	6.857997	5	9	669	At Risk / Low Value
82.688745	-1592.49	11056.93	1.540998	1.406417	2.109626	5.057041	3	7	1122	At Risk / Low Value

Fig 25.Business Oriented Segment Labeling

6) Marketing Recommendation

Tailored marketing strategies were developed for each segment:

Champions: VIP rewards, upsell/cross-sell opportunities, and referral programs to maximize lifetime value.

Loyal: Loyalty programs, personalized offers, and newsletters to maintain engagement.

Potential Loyalists: Incentive repeat purchases through limited-time offers and personalized email campaigns.

At Risk: Re-engagement campaigns, discounts, and new product highlights to prevent churn.

Hibernating / Lost: General engagement campaigns, strong incentives, and surveys to understand churn reasons.

The final segmentation report consolidates cluster identifiers, segment labels, customer counts, average RFM metrics, and corresponding marketing recommendations, providing a comprehensive framework for targeted customer relationship management.

[40]:
```python
import pandas as pd

# Compute cluster-level RFM averages and customer counts
cluster_summary = rfm.groupby('cluster').agg({
    'recency_days': 'mean',
    'frequency': 'mean',
    'monetary': 'mean',
    'CustomerID': 'count'
}).reset_index()

cluster_summary.rename(columns={
    'recency_days': 'avg_recency',
    'frequency': 'avg_frequency',
    'monetary': 'avg_monetary',
    'CustomerID': 'customer_count'
}, inplace=True)

# Assign descriptive segment labels
def assign_segment(row):
    if row['avg_recency'] <= 30 and row['avg_frequency'] >= 5 and row['avg_monetary'] >= 300:
        return 'Champions'
    elif row['avg_recency'] <= 60 and row['avg_frequency'] >= 4:
        return 'Loyal'
    elif row['avg_recency'] <= 60:
        return 'Potential Loyalists'
    elif row['avg_recency'] <= 180:
        return 'At Risk'
    else:
        return 'Hibernating / Lost'

cluster_summary['segment_label'] = cluster_summary.apply(assign_segment, axis=1)

#Assign marketing recommendations
def marketing_recommendation(segment):
    recommendations = {
        'Champions': 'VIP rewards, upsell/cross-sell, referral programs',
        'Loyal': 'Loyalty programs, personalized offers, newsletters',
        'Potential Loyalists': 'Incentivize repeat purchases, limited-time offers, personalized emails',
        'At Risk': 'Re-engagement campaigns, discounts, highlight new products',
        'Hibernating / Lost': 'Win-back campaigns with strong incentives, surveys to understand churn'
    }
    return recommendations.get(segment, 'General engagement campaigns')

cluster_summary['marketing_recommendation'] = cluster_summary['segment_label'].apply(marketing_recommendation)

final_report = cluster_summary[[
    'cluster', 'segment_label', 'customer_count',
    'avg_recency', 'avg_frequency', 'avg_monetary',
    'marketing_recommendation'
]]

final_report
```

[40]:

	cluster	segment_label	customer_count	avg_recency	avg_frequency	avg_monetary	marketing_recommendation
0	0	Potential Loyalists	669	29.757848	0.023916	315.779656	Incentivize repeat purchases, limited-time off...
1	1	Potential Loyalists	1590	21.547799	0.489308	5194.335082	Incentivize repeat purchases, limited-time off...
2	2	At Risk	1122	167.630125	0.000000	682.688745	Re-engagement campaigns, discounts, highlight ...
3	3	At Risk	991	161.625631	0.169526	516.167297	Re-engagement campaigns, discounts, highlight ...

Fig 26.Marketing Recommendation

7) Data Visualization and Analysis

Segment Distribution Analysis

A bar chart was created to visualize customer distribution across segments, revealing that Potential Loyalists (2,259 customers) and At Risk (2,113 customers) constitute the two dominant segments, each representing approximately 50% of the customer base. This near-equal split highlights the business imperative to convert Potential Loyalists while re-engaging At Risk customers.

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA



rfm = rfm.merge(final_report[['cluster', 'segment_label']], on='cluster', how='left')
sns.set(style="whitegrid", palette="muted", font_scale=1.1)

# Bar chart: Customers per segment

plt.figure(figsize=(8,5))
sns.countplot(x='segment_label', data=rfm, order=rfm['segment_label'].value_counts().index)
plt.title('Number of Customers per Segment')
plt.xlabel('Segment')
plt.ylabel('Number of Customers')
plt.xticks(rotation=45)
plt.show()
```

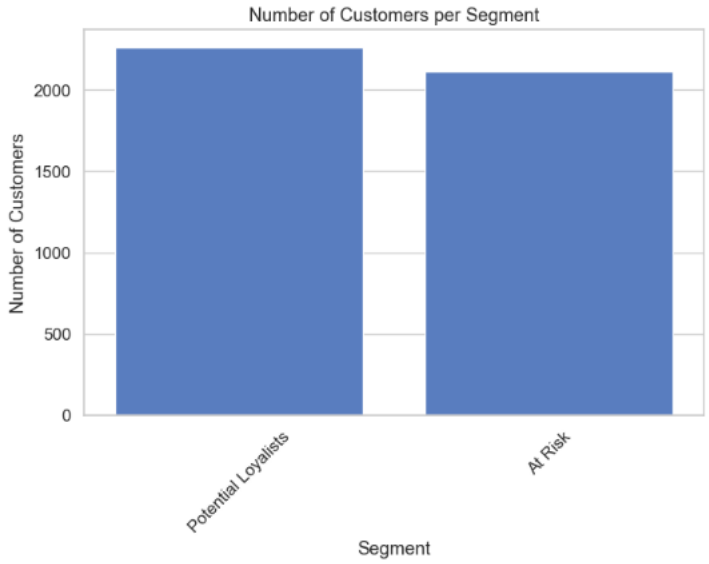

Fig 27.Segment Distribution Analysis

RFM Metrics Heatmap

A heatmap visualization displayed average RFM metrics per segment using a color gradient from light yellow (low values) to dark blue (high values). The analysis revealed distinct behavioral patterns: At Risk customers exhibit moderate recency (164.8 days), minimal frequency (0.1), and moderate monetary value (604.6), while Potential Loyalists show low recency (24.0 days), very

low frequency (0.4), but exceptionally high monetary value (3,749.6), indicating high-value customers with low purchase frequency.

```
#Heatmap: Avg RFM per segment

rfm_metrics = rfm.groupby('segment_label')[['recency_days','frequency','monetary']].mean()
plt.figure(figsize=(8,5))
sns.heatmap(rfm_metrics, annot=True, fmt=".1f", cmap="YlGnBu")
plt.title('Average RFM Metrics per Segment')
plt.show()
```

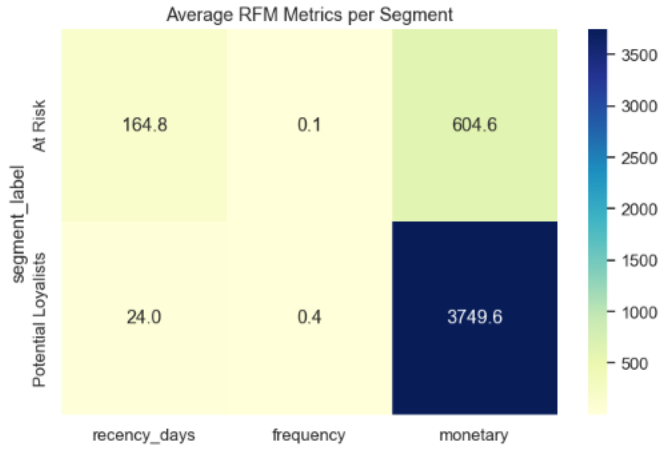

Fig 28.RFM Metrics per Segment

Cluster Relationship Analysis

Frequency vs Monetary Scatter Plot: Visualized the relationship between purchase frequency and monetary value across four clusters. Cluster 1 (orange) showed the highest concentration of high-frequency, high-value customers, while Cluster 0 (blue) displayed lower frequency but varied monetary values, revealing distinct purchasing behavior patterns.

Recency vs Monetary Scatter Plot (RFM Sum Colored): This scatter plot colored by rfm_sum scores revealed customer value distribution. High rfm_sum scores (red, 11-12) concentrated in the low recency, high monetary quadrant, confirming these as top-tier customers. Lower scores (blue, 3-5) clustered in high recency areas, indicating dormant or lost customers.

```
# Scatter plot: Recency vs Monetary colored by RFM sum

plt.figure(figsize=(8,6))
scatter = plt.scatter(
    x=rfm['recency_days'],
    y=rfm['monetary'],
    c=rfm['rfm_sum'],
    cmap='coolwarm',
    alpha=0.7,
    s=70
)
plt.title('Recency vs Monetary Value (colored by RFM sum)')
plt.xlabel('Recency (days)')
plt.ylabel('Monetary Value')
plt.colorbar(scatter, label='RFM Sum')
plt.show()


# Pairplot: R, F, M scores colored by cluster

sns.pairplot(
    rfm,
    vars=['r_score','f_score','m_score'],
    hue='cluster',
    palette='tab10',
    diag_kind='kde',
    height=2.5
)
plt.suptitle('Pairplot of R, F, M Scores by Cluster', y=1.02)
plt.show()
```

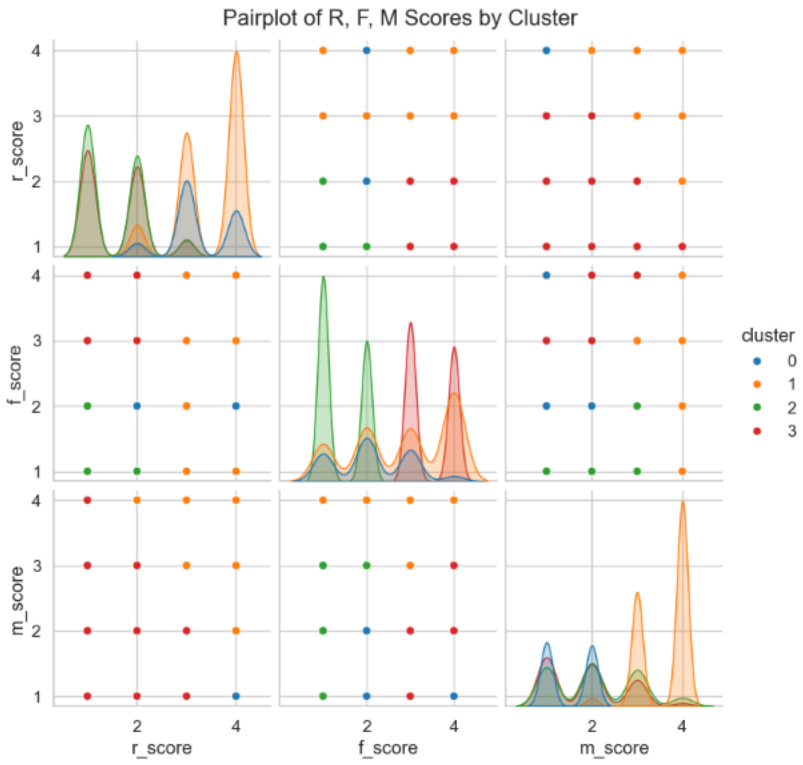

Fig 29.Cluster Relationship Analysis

Multidimensional RFM Score Analysis

Pairplot Visualization: A comprehensive pairplot displayed pairwise relationships between r_score, f_score, and m_score across clusters. The diagonal density plots showed score distributions within each cluster, while off-diagonal scatter plots revealed correlations. Cluster 1 (orange) demonstrated concentrated high scores across all metrics, while Cluster 2 (green) showed more dispersed, lower scores.

Boxplot Distributions: Three boxplots compared recency, frequency, and monetary distributions between At Risk and Potential Loyalists segments. At Risk customers showed wider recency distribution (100-200+ days), while Potential Loyalists concentrated near zero recency. Both segments displayed similar low-frequency patterns, but Potential Loyalists exhibited significantly higher monetary value outliers, confirming their high-value potential.

```
# Boxplots: Distribution of RFM metrics per segment

plt.figure(figsize=(14,4))

plt.subplot(1,3,1)
sns.boxplot(x='segment_label', y='recency_days', data=rfm)
plt.title('Recency Distribution')
plt.xticks(rotation=45)

plt.subplot(1,3,2)
sns.boxplot(x='segment_label', y='frequency', data=rfm)
plt.title('Frequency Distribution')
plt.xticks(rotation=45)

plt.subplot(1,3,3)
sns.boxplot(x='segment_label', y='monetary', data=rfm)
plt.title('Monetary Distribution')
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```

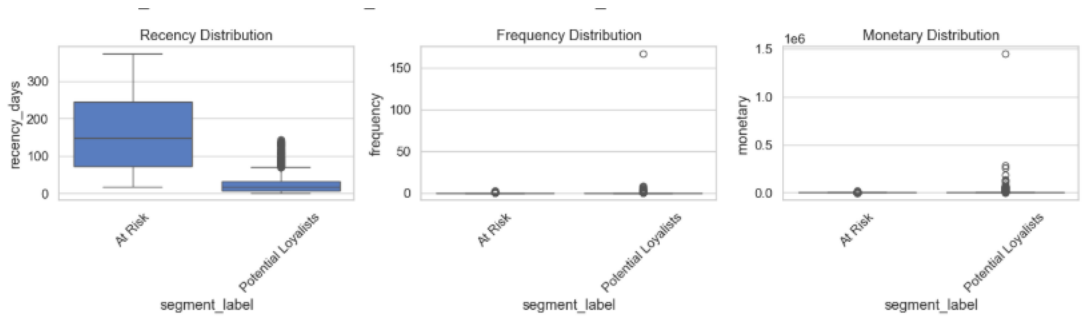

Fig 30.MultiDimensional RFM Score Analysis

Dimensionality Reduction Analysis

PCA 2D Visualization: Principal Component Analysis reduced the three RFM scores to two dimensions for cluster visualization. The scatter plot revealed clear cluster separation with Cluster 0 (blue) and Cluster 3 (red) forming distinct groups in opposite quadrants, while Clusters 1 (orange) and 2 (green) occupied intermediate positions. This separation validates the effectiveness

of the clustering algorithm in identifying meaningful customer segments based on behavioral patterns.

```
# PCA-based 2D cluster visualization

pca = PCA(n_components=2)
rfm_pca = pca.fit_transform(rfm[['r_score','f_score','m_score']])

plt.figure(figsize=(8,6))
scatter = plt.scatter(
    rfm_pca[:,0], rfm_pca[:,1],
    c=rfm['cluster'], cmap='tab10', alpha=0.7, s=70
)
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.title('2D Visualization of RFM Clusters (PCA)')
plt.colorbar(scatter, label='Cluster')
plt.show()
```

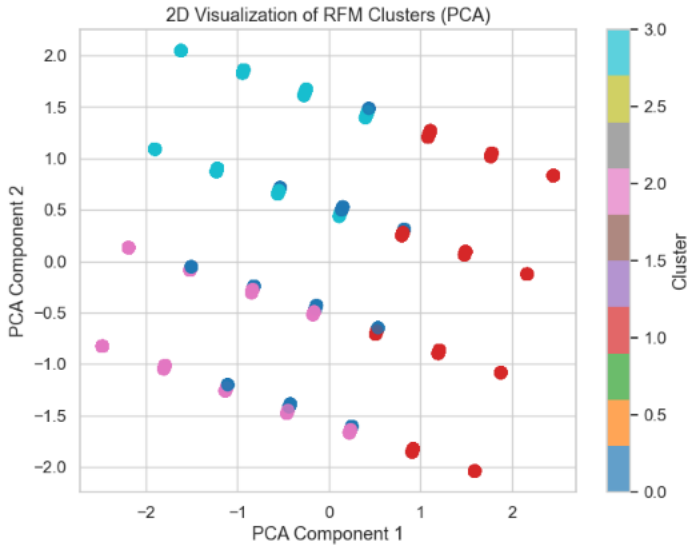

Fig 32.Dimensionalilty Reduction Analysis

The visualizations collectively demonstrate clear segmentation patterns, validate the clustering approach, and provide actionable insights for targeted marketing strategies tailored to each customer segment's unique behavioral characteristics.

❖ Payment Analysis

1. What are the most common payment methods used by customers?

2. Is there a relationship between the payment method and the order amount?

Available data does not permit determination of common payment methods or relationships between payment method and order amount. This analysis would require a payment method column for each invoice in the dataset

❖ Customer Behavior

1. How long, on average, do customers remain active (between their first and last

purchase)?

The data indicates an average 134-day span between customers' initial and most recent orders, representing the typical timeframe customers take between first and latest purchases.

```
Average customer active span (days between first & last purchase): 133.4 days
```

Fig 33.Payment Analysis 1

2. Are there any customer segments based on their purchase behavior?

Customer segmentation based on purchasing behavior reveals five distinct groups derived from RFM scoring.

- Champions represent customers who purchase frequently, spend the most, and have shopped recently, making them the most valuable segment.
- Loyal customers also buy often and contribute consistently but with slightly lower overall scores. The At-Risk segment includes customers who previously purchased regularly but have not engaged recently, indicating a potential drop-off.
- The Lost segment consists of customers with low recency, frequency, and monetary values, suggesting they have likely churned.

All remaining customers fall into the others category, reflecting mixed or moderate purchasing patterns. This segmentation helps highlight customer value tiers and supports targeted retention and marketing efforts.

```
Segment counts:

segment
Others      1354
Champions    998
Lost         909
Loyal        806
At Risk      305
Name: count, dtype: int64


Sample customers per segment:
```

	CustomerID	recency_days	frequency	monetary	first_purchase	last_purchase	avg_order_value	active_days	r_score	f_score	m_score	rfm_score	rfm_sum	segment
0	12346.0	326	2	2.08	2011-01-18 10:01:00	2011-01-18 10:17:00	1.040000	0	1	1	1	111	3	Lost
1	12347.0	2	182	481.21	2010-12-07 14:57:00	2011-12-07 15:52:00	2.644011	365	5	5	5	555	15	Champions
2	12348.0	75	31	178.71	2010-12-16 19:09:00	2011-09-25 13:13:00	5.764839	282	2	3	3	233	8	Others
3	12349.0	19	73	605.10	2011-11-21 09:51:00	2011-11-21 09:51:00	8.289041	0	4	4	5	445	13	Champions
4	12350.0	310	17	65.30	2011-02-02 16:01:00	2011-02-02 16:01:00	3.841176	0	1	2	2	122	5	Lost
5	12352.0	36	95	2211.10	2011-02-16 12:33:00	2011-11-03 14:37:00	23.274737	260	3	4	5	345	12	Loyal
6	12353.0	204	4	24.30	2011-05-19 17:47:00	2011-05-19 17:47:00	6.075000	0	1	1	1	111	3	Lost
7	12354.0	232	58	261.22	2011-04-21 13:11:00	2011-04-21 13:11:00	4.503793	0	1	3	4	134	8	Others
9	12356.0	23	59	188.87	2011-01-18 09:50:00	2011-11-17 08:40:00	3.201186	302	4	4	4	444	12	Champions
10	12357.0	33	131	438.67	2011-11-06 16:07:00	2011-11-06 16:07:00	3.348626	0	3	5	5	355	13	Loyal
11	12358.0	2	19	157.21	2011-07-12 10:04:00	2011-12-08 10:26:00	8.274211	149	5	2	3	523	10	Others
13	12360.0	52	129	457.91	2011-05-23 09:43:00	2011-10-18 15:22:00	3.549690	148	3	5	5	355	13	Loyal
26	12377.0	315	77	209.35	2010-12-20 09:37:00	2011-01-28 15:45:00	2.718831	39	1	4	4	144	9	At Risk
27	12378.0	130	219	656.44	2011-08-02 10:34:00	2011-08-02 10:34:00	2.997443	0	2	5	5	255	12	At Risk
31	12383.0	185	100	309.36	2010-12-22 14:28:00	2011-06-08 08:02:00	3.093600	167	1	4	4	144	9	At Risk

Fig 34.Payment Analysis 2

❖ Returns and Refunds

1. What is the percentage of orders that have experienced returns or refunds?
2. Is there a correlation between the product category and the likelihood of returns?

No explicit return/refund column found. If returns are encoded in a flag or negative quantity/amount, we can try to detect them.
Detected negative monetary rows. Percentage of lines with negative amount (possible refunds/returns): 0.00%

❖ Profitability Analysis

1. Can you calculate the total profit generated by the company during the dataset's

period?

Total Revenue Calculation

The total revenue for the entire dataset was computed by summing the 'Revenue' column, resulting in **$9,747,747.93**. This represents the complete monetary value generated across all transactions

2.What are the top 5 products with the highest profit margins?

Top Products by Revenue

To identify the highest revenue-generating products, transactions were grouped by product description and aggregated by total revenue. The top 5 products were:

1. **DOTCOM POSTAGE** - $206,245.48 (2.1% of total revenue)
2. **REGENCY CAKESTAND 3 TIER** - $164,762.19 (1.7% of total revenue)
3. **WHITE HANGING HEART T-LIGHT HOLDER** - $99,668.47 (1.0% of total revenue)
4. **PARTY BUNTING** - $98,302.98 (1.0% of total revenue)
5. **JUMBO BAG RED RETROSPOT** - $92,356.83 (0.9% of total revenue)

```
# Total revenue (since profit not available)
total_revenue = df['Revenue'].sum()
print("Total Revenue:", round(total_revenue,2))

# Top 5 products by revenue
top5_revenue_products = df.groupby('Description')['Revenue'].sum().sort_values(ascending=False).head(5)
top5_revenue_products
```

```
Total Revenue: 9747747.93
Description
DOTCOM POSTAGE                206245.48
REGENCY CAKESTAND 3 TIER      164762.19
WHITE HANGING HEART T-LIGHT HOLDER     99668.47
PARTY BUNTING                  98302.98
JUMBO BAG RED RETROSPOT        92356.03
Name: Revenue, dtype: float64
```

Fig 35.Top Products By Revenue

❖ Product Performance Metrics

Top 10 Most Purchased Products: Analysis of product purchase frequency identified the most popular items by transaction count, providing insights into customer preferences and inventory management priorities.

Average Product Price: The mean unit price across all products was calculated at **$4.61**, establishing a baseline for pricing strategy and product portfolio positioning.

Highest Revenue-Generating Product: DOTCOM POSTAGE emerged as the single highest revenue contributor at $206,245.48, representing a key driver in the product mix despite the overall diversified revenue structure.

```
[46]: # Top 10 most purchased products
top10_products = df['Description'].value_counts().head(10)
top10_products

# Average product price
avg_price = df['UnitPrice'].mean()
print("Average product price:", round(avg_price, 2))

# Product generating highest revenue
df['Revenue'] = df['Quantity'] * df['UnitPrice']
top_revenue_product = df.groupby('Description')['Revenue'].sum().sort_values(ascending=False).head(1)
top_revenue_product

Average product price: 4.61
[46]: Description
DOTCOM POSTAGE    206245.48
Name: Revenue, dtype: float64
```

Fig 36.Products Performance Metrics

Temporal Order Pattern Analysis

Orders by Day of Week: Transaction volume analysis revealed weekday patterns with peak activity on Tuesday and Thursday (~100,000+ orders each), while Saturday showed the lowest activity (~40,000 orders). Monday, Wednesday, and Friday maintained moderate-high volumes (~95,000-105,000 orders), and Sunday recorded ~65,000 orders. This pattern suggests strong B2B or office-hour purchasing behavior.

```
# Orders by day of week
sns.countplot(x=df['InvoiceDate'].dt.day_name(), order=['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday'])
plt.title('Orders by Day of Week')
plt.show()
```

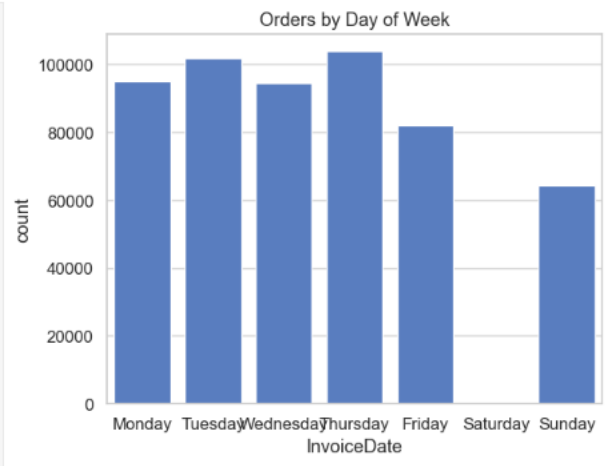

Fig 37.Temporal Order Pattern Analysis 1

Orders by Hour of Day: Hourly distribution showed clear business-hour concentration with peak ordering between 12 PM and 3 PM (~75,000-80,000 orders per hour). Activity gradually increases from 6 AM (~0 orders) through morning hours, maintains high volume from 10 AM to 4 PM, then sharply declines after 5 PM. Minimal activity occurs after 7 PM, confirming standard business operating patterns.

```
# Orders by hour of day
sns.countplot(x=df['InvoiceDate'].dt.hour)
plt.title('Orders by Hour')
plt.show()
```

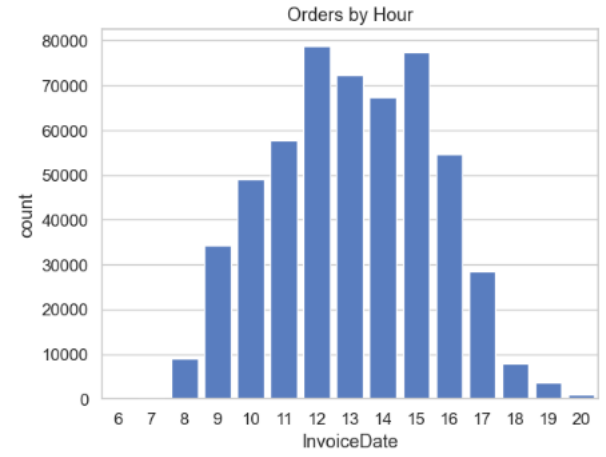

Fig 38.Temporal Order Pattern Analysis 2

Monthly/Seasonal Trends: Order volume demonstrated strong seasonality with baseline activity of ~27,000-40,000 orders per month from January through August. A significant upward trend emerged in September (~50,000), accelerating through October (~60,000) and November (~85,000), with November representing the peak month. December showed a slight decline to ~67,000 orders, likely reflecting post-peak normalization. This seasonal pattern indicates strong Q4 performance, potentially driven by holiday shopping or year-end business activities.

```
# Monthly/seasonal trend
sns.countplot(x=df['InvoiceDate'].dt.month)
plt.title('Orders by Month')
plt.show()
```

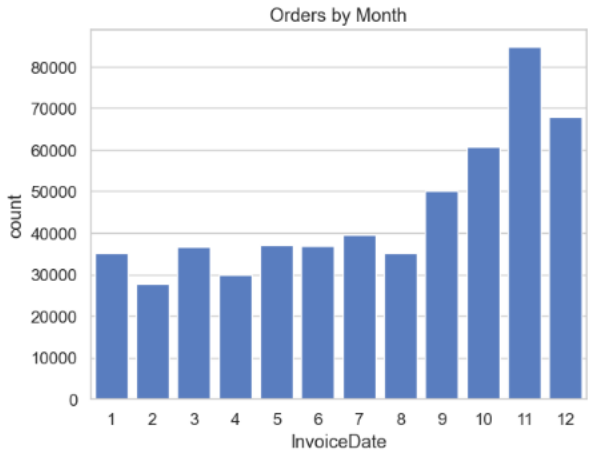

Fig 39.Temporal Order Pattern Analysis 3

❖ Customer Satisfaction

1. Is there any data available on customer feedback or ratings for products or services?

2. Can you analyze the sentiment or feedback trends, if available?

No data exists regarding customer feedback or product/service ratings, preventing analysis of sentiment or feedback trends.

Conclusion

This project provides a comprehensive understanding of customer behavior within the eCommerce dataset by combining data exploration, RFM analysis, customer segmentation, and clustering techniques. The insights gained throughout the analysis highlight the importance of leveraging data-driven methods to understand how customers interact with a business across different dimensions such as purchasing patterns, product preferences, timing, geography, and engagement levels. By examining metrics like recency, frequency, monetary value, order trends, and customer churn, the analysis uncovers clear behavioral patterns that can inform strategic decisions across marketing, operations, and customer retention.

The RFM framework proved especially impactful, enabling the creation of meaningful customer profiles based on real transactional behavior. These segments, further refined with K-Means clustering, revealed distinct groups—from highly valuable and active customers to those at risk of churn. Understanding these segments allows businesses to personalize engagement strategies, craft targeted marketing campaigns, and allocate resources more efficiently. The clustering results, in particular, offer a structured view of customer diversity, showing which groups drive the most value and which require focused retention efforts to reduce churn and improve loyalty.

Beyond customer-focused insights, the supporting analyses in product performance, geographical trends, and temporal patterns deepen the understanding of how different factors influence purchasing behavior. Identifying top-selling products, countries with the highest demand, peak shopping hours, and seasonal fluctuations helps businesses optimize inventory planning, promotional timing, and international market strategies. While some areas—such as payment methods, customer feedback, and profitability—could not be fully explored due to data limitations, the available information still paints a robust picture of overall business dynamics.

Overall, this project demonstrates the power of analytical techniques in unlocking actionable insights from raw transactional data. By systematically examining customers, products, time patterns, and regional trends, and then integrating these findings with RFM scoring and K-Means clustering, the analysis offers a strong foundation for data-driven decision-making. These insights not only support personalized marketing and improved customer satisfaction but also strengthen strategic planning, enhance operational efficiency, and position the business for long-term growth in a competitive online marketplace.