UNIVERSITY OF WESTMINSTER

7BUIS010W.2

Data Warehousing and Business Intelligence

Assessment 02

April 2024

Module Leader:
Panagiotis Chountas

Akram Naoufel Tabet
w1979246

1. Project Overview and Objectives

a. Overview

This project is designed to harness the power of data analytics in understanding customer behaviours through transactional data. The primary focus is to employ RFM segmentation and DBSCAN clustering methodologies to uncover distinct customer segments, each characterised by specific purchasing patterns. This strategic insight aims to empower the marketing department to tailor their campaigns more effectively and efficiently, enhancing customer engagement and boosting retention rates.

b. Objectives

1. **Perform RFM Segmentation:** To categorise customers based on Recency, Frequency, and Monetary values, providing a foundational understanding of the customer base.
2. **Apply DBSCAN Clustering:** To identify naturally occurring groups within the customer data, offering a nuanced view of customer behaviours and preferences.
3. **Enhance Marketing Strategies:** To utilise the insights gained from RFM segmentation and DBSCAN clustering to develop targeted marketing actions that improve customer loyalty and increase customer lifetime value.
4. **Design a Data Mart:** To structure a scalable and efficient data mart that supports the ongoing analysis needs of the marketing department, enabling quick access to key metrics and dimensions that drive decision-making.

c. Dataset Description

Data Overview

The dataset used in this project comprises transaction records from a retail environment, capturing customer purchases over a specific period. Each record details the customer's interaction with the product offerings, providing a rich source of information for analysis.

Attributes

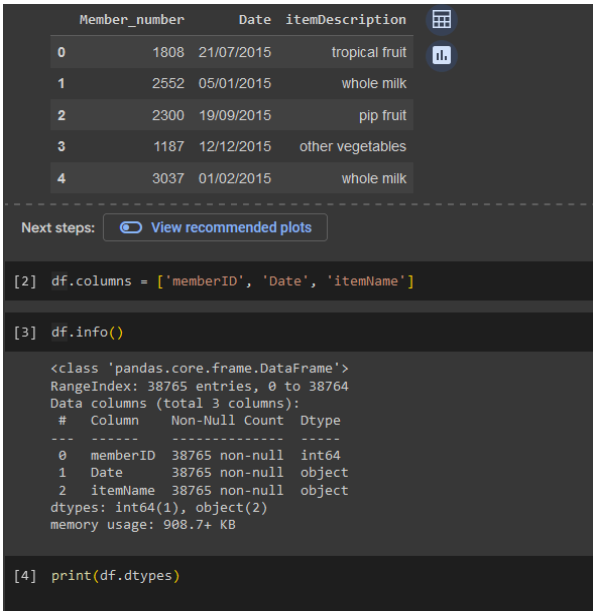
- **Member Number:** A unique identifier for each customer.
- **Date:** The date on which each transaction occurred.
- **Item Description:** Describes the product purchased in each transaction.

Size and Scope

- **Total Records:** 38,765 entries, representing individual purchase actions.
- **Coverage:** The dataset spans multiple product categories and includes data from various geographic locations, offering a comprehensive view of the customer base.

Data Understanding

Before initiating the data distribution analysis, it was essential to preprocess and clean the dataset to ensure accuracy in the subsequent steps. The process began with a thorough examination of the dataset, including an overview of column names, data types, and the count of non-null values. To simplify the data handling, I renamed the 'Member_Number' and 'ItemDescription' columns to 'memberID' and 'itemName', respectively.



	Member_number	Date	itemDescription
0	1808	21/07/2015	tropical fruit
1	2552	05/01/2015	whole milk
2	2300	19/09/2015	pip fruit
3	1187	12/12/2015	other vegetables
4	3037	01/02/2015	whole milk

Next steps: View recommended plots

```
[2] df.columns = ['memberID', 'Date', 'itemName']

[3] df.info()

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 38765 entries, 0 to 38764
    Data columns (total 3 columns):
     #   Column    Non-Null Count  Dtype
    ---  ------    --------------  -----
     0   memberID  38765 non-null  int64
     1   Date      38765 non-null  object
     2   itemName  38765 non-null  object
    dtypes: int64(1), object(2)
    memory usage: 908.7+ KB

[4] print(df.dtypes)
```

Figure 01. Initial Overview of the dataset

Management of missing values and suitable transformations of variables:

An initial check for missing values revealed that there were none, as indicated in the attached figure. Additionally, data types for certain columns were converted for appropriate analysis: the 'Date' column was changed from an object type to a date format, and 'itemName' was converted to a string format, as shown in the figures.

```
[5] # Check for missing values in the dataset
    missing_values = df.isnull().sum()

    print("Missing Values:\n")
    print(missing_values)

    Missing Values:

    memberID    0
    Date        0
    itemName    0
    dtype: int64

[28] # Convert 'Date' column to datetime format
    df['Date'] = pd.to_datetime(df['Date'], format='%d/%m/%Y')

    # Convert 'itemDescription' column to string type
    df['itemName'] = df['itemName'].astype('string')
    print(df.dtypes)

    memberID             int64
    Date        datetime64[ns]
    itemName    string[python]
    dtype: object
```

Figure 02. Management of missing values and transformations of variables

Elimination of redundant variables:

To ensure data integrity, I examined the dataset for unique values in 'itemName' to identify any inappropriate or erroneous entries that needed removal. Furthermore, I checked for duplicate transactions, a common issue in transactional data, and confirmed their presence as expected.

```
df[df.duplicated()].sort_values('memberID')
```

	memberID	Date	itemName
37649	1003	2014-02-27	rolls/buns
31248	1005	2014-01-09	rolls/buns
8488	1006	2015-06-14	frankfurter
21210	1008	2015-07-21	yogurt
20918	1026	2015-05-28	root vegetables
...	...	...	...
24876	4968	2014-03-04	beef
18312	4981	2015-01-10	margarine
24043	4981	2015-01-10	margarine
24258	4988	2015-10-29	rolls/buns
38136	4992	2014-02-24	margarine

759 rows × 3 columns

```
[25] df.drop_duplicates(inplace=True)
```

Figure 03. Dropping duplicate values

In the process of data wrangling, several tasks were undertaken: managing missing values, transforming suitable variables, and eliminating redundant data. Specifically, I identified and removed approximately 759 redundant transactions, reducing the dataset to 38,006 transactions. These adjustments ensured that the dataset was clean and well-prepared for distribution analysis and further processing.

Distribution Analysis and statistical exploration with visualisation

The initial exploration of the dataset unveiled key distribution properties of the transactional data, providing a foundation for deeper analysis. To assess the distribution, I conducted value counts across various columns, which helped determine the volume of transactions per member, the top purchased products, and the days most likely to see transactions. This analysis sets the stage for more detailed visual evaluations, including:

- The frequency distribution of item purchases.
- The number of items purchased per member.
- The distribution of transactions over time.

In the realm of statistical exploration, I performed basic statistics on 'memberID' and 'itemName'. This analysis provided insights into the most frequently purchased items, the number of unique items, and standard deviation among other statistics for these categorical variables. Additionally, I assessed numerical variables to derive basic statistical metrics. All these activities were underpinned by theoretical approaches and actual coding, as demonstrated in the figures.

```python
# Distribution analysis for Member_number
member_number_counts = df['memberID'].value_counts()
print("Distribution analysis for Member_number:")
print(member_number_counts)

# Distribution analysis for itemDescription
item_description_counts = df['itemName'].value_counts()
print("\nDistribution analysis for itemDescription:")
print(item_description_counts)

# Distribution analysis for Date
date_counts = df['Date'].value_counts()
print("\nDistribution analysis for Date:")
print(date_counts)
```

```
Distribution analysis for Member_number:
memberID
3180    35
3737    33
3050    32
2051    31
3915    30
        ..
4816     1
4029     1
4151     1
4565     1
2640     1
Name: count, Length: 3898, dtype: int64

Distribution analysis for itemDescription:
itemName
whole milk              2363
other vegetables        1827
rolls/buns              1646
soda                    1453
yogurt                  1285
                        ...
rubbing alcohol            5
bags                       4
baby cosmetics             3
```

```python
# Calculate basic statistics for the numerical data
numerical_stats = df['memberID'].describe()

# Categorical data statistics for 'itemName'
categorical_stats = df['itemName'].describe()

numerical_stats, categorical_stats
```

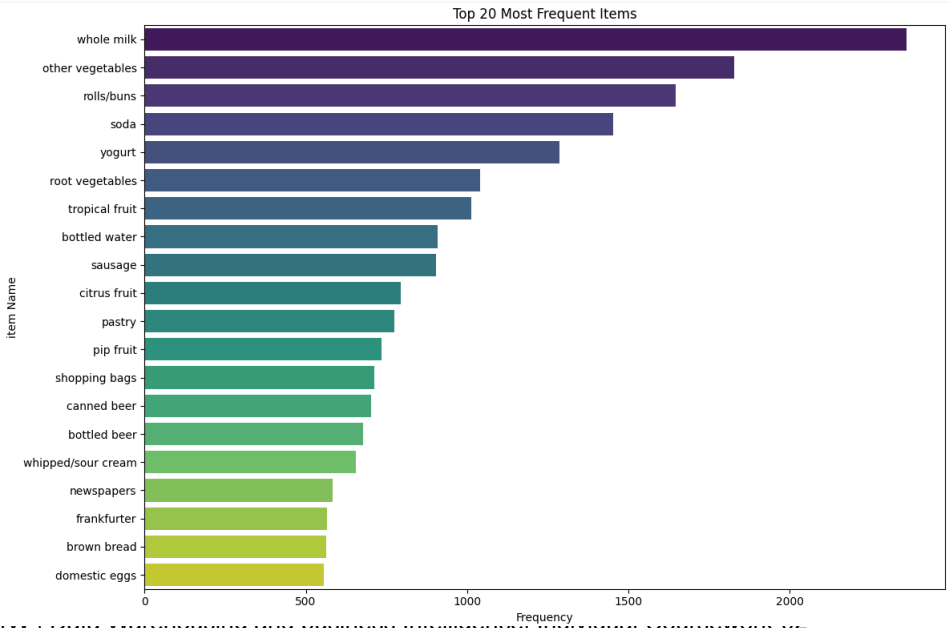
```
(count    38006.000000
 mean      3003.361390
 std       1153.659151
 min       1000.000000
 25%       2001.000000
 50%       3005.000000
 75%       4007.000000
 max       5000.000000
 Name: memberID, dtype: float64,
 count          38006
 unique           167
 top       whole milk
 freq            2363
 Name: itemName, dtype: object)
```

Figure 04 . Distribution Analysis and statistics

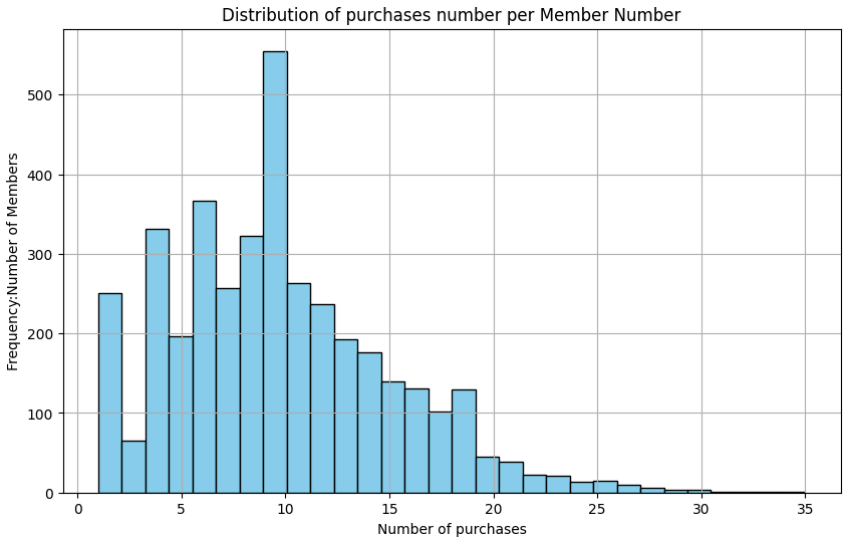
Now, let's progress to the most critical phase 'Data visualisation' where these statistical insights will be transformed into graphical formats for more intuitive understanding and analysis.

Data visualisation

Here is a bar chart displaying the top 20 most purchased items from our dataset. This visualisation helps in understanding which items are more frequently bought by the members.



Distribution of purchased items per member



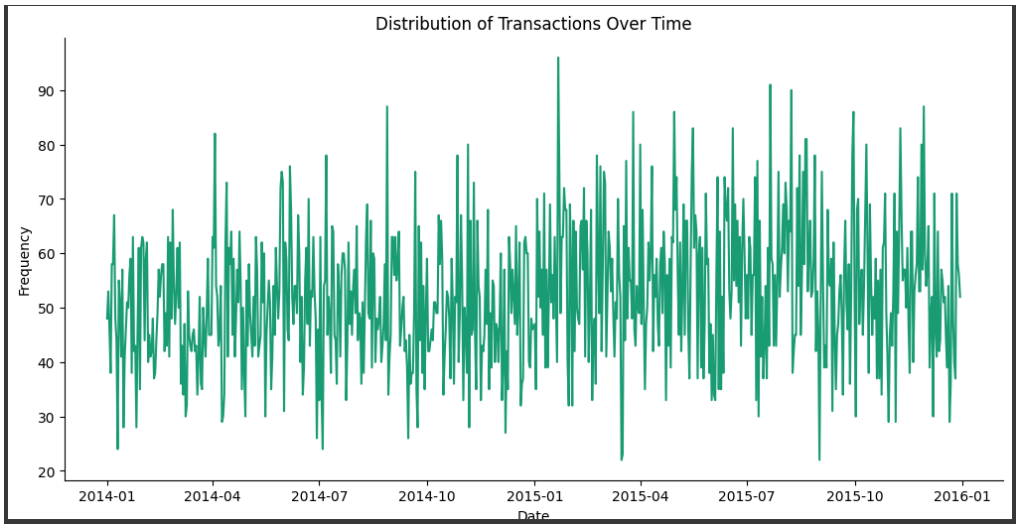
The histogram shows the distribution of the number of purchases per member, with some statistics to provide further insights:

- **Mean**: On average, each member made about 9.75 purchases.
- **Standard Deviation**: The standard deviation is*slightly lower at 5.19, indicating a bit less variability in shopping frequency among members.
- **Minimum**: The minimum number of purchases by a member is 1.
- **25th Percentile (Q1)**: 25% of the members made 6 or fewer purchases.
- **Median (50th Percentile)**: The median remains at 9 purchases, indicating that half of the members made up to 9 purchases.
- **75th Percentile (Q3)**: 75% of the members made 13 or fewer purchases.
- **Maximum**: The maximum number of purchases by a member is 35.

Observations from the Updated Histogram:

- The distribution is still right-skewed, indicating that a significant number of members make purchases more frequently than the average.
- The majority of members still fall within the 6 to 13 purchase range.
- There are few members with extremely high purchase counts, up to 35, which appear as outliers.

The distribution of transactions over time:



Here is the line graph showing the distribution of transactions over time. This visualisation helps us observe trends, seasonality, and any significant spikes or drops in transaction activity.

Observations from the Graph:

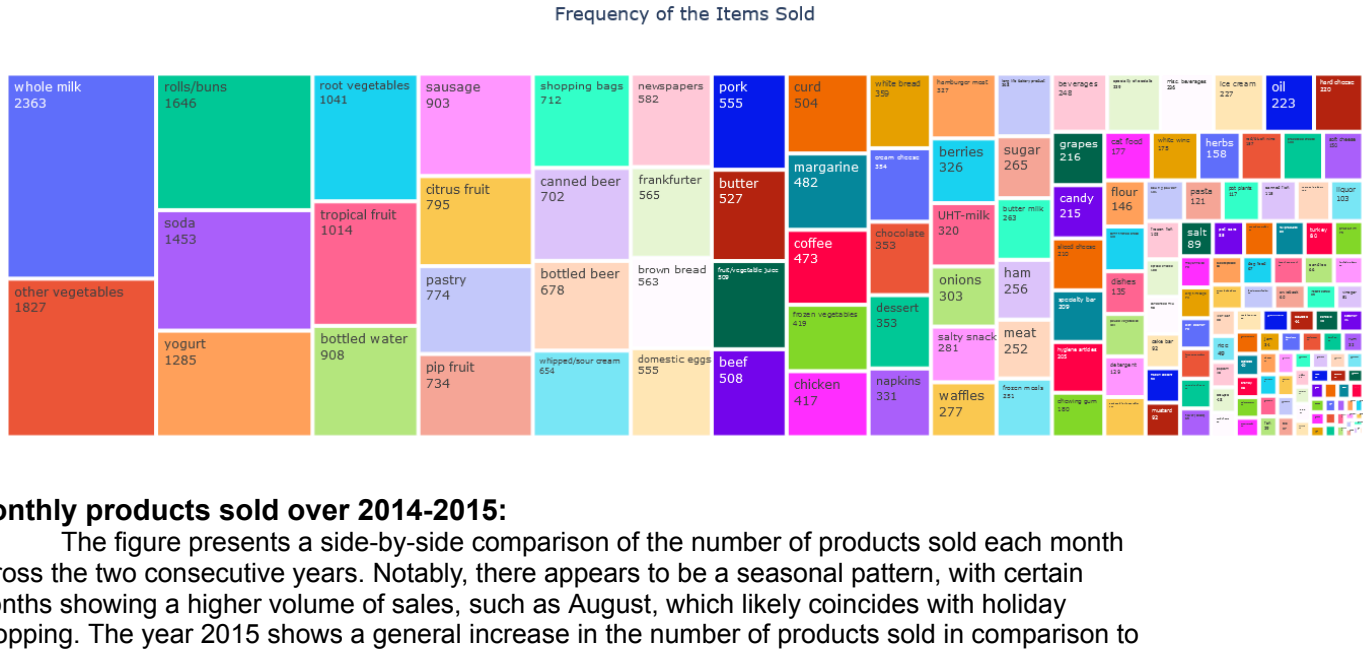
- There are fluctuations in the number of transactions per day throughout the period covered by the data.
- Some peaks suggest days with unusually high transaction activity, which could correspond to special events, promotions, or seasonal buying patterns.
- Similarly, the dips might indicate days with low activity, such as holidays or off-season periods.

The distribution of customers by quantity purchased

It is immediately noticeable that for most customers, the quantity purchased in 2015 exceeds that of 2014, which may suggest increased customer retention and loyalty, or possibly an expansion in product offerings that has resonated with these customers.

The number of items purchased per member

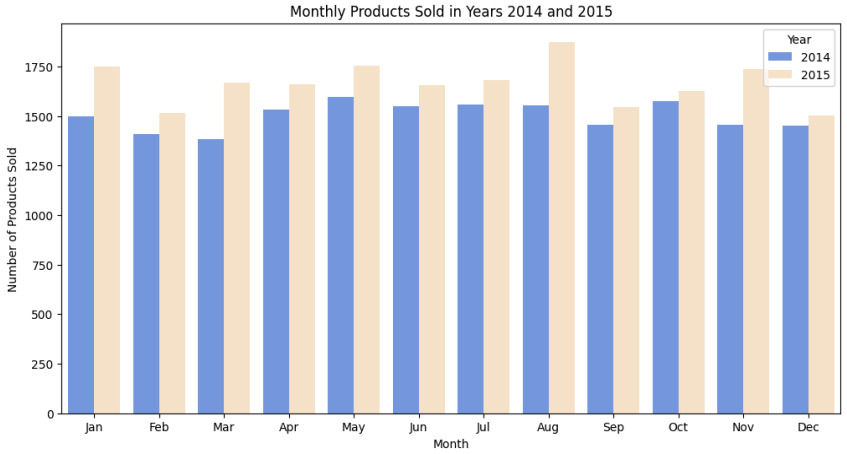
This tree map represents the sales frequency of various items, with the most popular items like 'whole milk' and 'other vegetables' occupying prominently larger blocks.



Monthly products sold over 2014-2015:

The figure presents a side-by-side comparison of the number of products sold each month across the two consecutive years. Notably, there appears to be a seasonal pattern, with certain months showing a higher volume of sales, such as August, which likely coincides with holiday shopping. The year 2015 shows a general increase in the number of products sold in comparison to

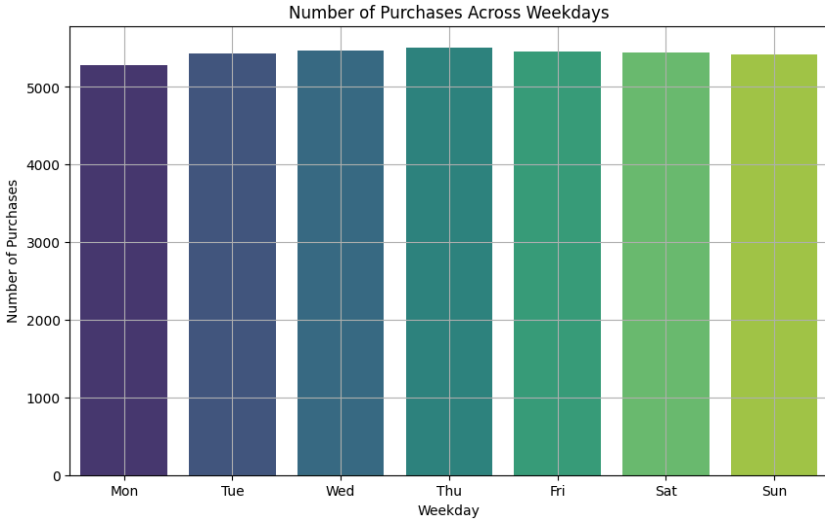
2014, suggesting a positive trend in sales volume over the year. This could be indicative of successful



marketing strategies.

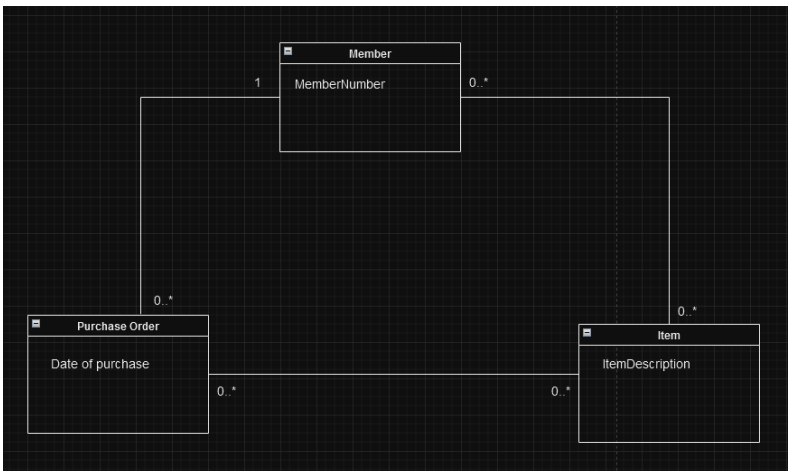
Number of Purchases Across Weekdays

In the second chart, we observe the distribution of purchase transactions across the days of the week. The pattern exhibited shows that Thursday is the busiest shopping day, with the number of purchases peaking noticeably. The least number of purchases occur on Monday, which may be an opportunity to implement special promotions or marketing efforts to increase traffic on this underperforming day.



I've created a range of data visualisations; however, due to the page limit of 15 for reports, I was unable to include all of them. Instead, I've selected and presented the most compelling visualisations that offer valuable information and insights, ensuring that we focus on the data that provides the greatest opportunity for in-depth analysis and understanding.

UML Class diagram:



In this section of the project, I've delineated the structure of our data model using a UML class diagram to illustrate the relationships between various entities within the transactional database. The diagram highlights three primary classes: 'Member', 'Purchase Order', and 'Item'.

Class Identification:

- The 'Member' class is characterised by a 'MemberNumber', which uniquely identifies each member within the system.
- The 'Purchase Order' class encapsulates the 'Date of purchase', detailing when transactions occur.
- The 'Item' class is described by an 'ItemDescription', providing information on each product sold.

Association Identification:

- 'Member' and 'Purchase Order' are associated, as members create purchase orders.
- 'Purchase Order' and 'Item' have an association, indicating that each purchase order includes one or more items.

Cardinality Constraints:

- A 'Member' can be associated with zero or many 'Purchase Orders', indicating that members may have multiple transactions or none at all within the dataset's timeframe.
- Each 'Purchase Order' is connected to one 'Member', signifying that a purchase is made by an individual member.
- A 'Purchase Order' can contain zero or many 'Items', reflecting the fact that a transaction may include multiple items or possibly none if it's a cancellation or return.
- Similarly, each 'Item' can be part of zero or many 'Purchase Orders', which accounts for the fact that any particular item can be bought across multiple transactions.

RFM Segmentation in sql:

RFM is a method used for analysing customer value. It is commonly used in database marketing and direct marketing and has received particular attention in the retail and professional services industries.

RFM metrics:

Direct marketers use three readily available customer behaviours:

Recency or time elapsed since last purchase:(When was the last time your customer purchased a product/service?) A high recency score means a customer has positively considered your brand for a purchase decision recently.

Frequency of purchases in the last period: (How often did the customer purchase in a year/fixed time period?) A high-frequency score means a customer buys your brand frequently and is likely to be a loyalist of your brand.

Monetary purchases in the last period: (How much money has the customer spent on your brand so far?) A high monetary value score means a customer is one of the highest spending customers of your brand.

These RFM (recency, frequency, and monetary) variables put customers in rank-ordered groups, based on their value in the past year (not by modelling but by rank-order sorting)

Implementation of the RFM metrics in python using sqlite:

As part of the project, an RFM (Recency, Frequency, Monetary) model has been developed to evaluate customer value. The SQL query provided effectively assigns each member with these critical values based on their transaction history. The 'Recency' column denotes the number of days since the last purchase, offering insight into customer engagement levels. 'Frequency' reflects the number of distinct shopping days, indicating customer loyalty and the habit of repeat purchases. 'Monetary' represents the total number of transactions, which serves as a proxy for the money spent by each member. Here is the query and the code implemented:

```
[65] import sqlite3

    # Connect to an in-memory SQLite database
    conn = sqlite3.connect(':memory:')

    df.to_sql('transactions', conn, index=False, if_exists='replace', dtype={
        'memberID': 'INTEGER',
        'Date': 'TEXT',
        'itemName': 'TEXT'
    })

    rfm_query = """
    WITH LastPurchaseDate AS (
        SELECT MAX(julianday(Date)) as MaxDate FROM transactions
    )
    SELECT
        t.memberID AS CustomerID,
        MAX(Date) AS LastPurchaseDate,
        CAST((SELECT MaxDate FROM LastPurchaseDate) - MAX(julianday(t.Date)) AS INTEGER) AS Recency, -- Days since last purchase from the last date in dataset
        COUNT(DISTINCT t.Date) AS Frequency, -- Visit frequency: distinct days with purchases
        COUNT(*) AS Monetary -- Total number of transactions
    FROM
        transactions t
    GROUP BY
        t.memberID
    ORDER BY
        t.memberID;


    """

    rfm_results = pd.read_sql_query(rfm_query, conn)

    rfm_results
```

RFM metrics Table:

	CustomerID	LastPurchaseDate	Recency	Frequency	Monetary
0	1000	2015-11-25 00:00:00	35	5	13
1	1001	2015-05-02 00:00:00	242	5	12
2	1002	2015-08-30 00:00:00	122	4	8
3	1003	2015-02-10 00:00:00	323	4	7
4	1004	2015-12-02 00:00:00	28	8	21
...	...	...	...	...	...
3893	4996	2015-11-24 00:00:00	36	3	10
3894	4997	2015-12-27 00:00:00	3	2	6
3895	4998	2015-10-14 00:00:00	77	1	2
3896	4999	2015-12-26 00:00:00	4	6	16
3897	5000	2015-02-10 00:00:00	323	3	7

3898 rows × 5 columns

The table is a result of the RFM segmentation model executed via SQL, which assigns each customer with Recency, Frequency, and Monetary values based on their purchase history. This table includes:

- **LastPurchaseDate:** The date of the most recent purchase made by the customer.
- **Recency:** The number of days between the last purchase date in the dataset and the last purchase date for each customer. It indicates how recently each customer has made a purchase.
- **Frequency:** The number of distinct shopping days, signifying how often each customer shops.
- **Monetary:** The total number of transactions made by each customer, which can be a proxy for the total spend.

This RFM table provides a comprehensive snapshot of customer engagement and purchasing behaviour, serving as a foundational tool for personalised marketing strategies, customer lifecycle analysis, and tailored customer engagement programs. The recency and frequency information can help identify loyal customers, while the monetary value helps in recognizing the top spenders.

I have created a visualisation of the distribution for each column(recency,frequency and monetary) but due to the page limit of 15 for reports, I was unable to include all of them. It's well presented on the jupiter file.

Customer segmentation with DBSCAN:

Data Standardization

Prior to the development of the DBSCAN model, it is crucial to standardise the dataset. This preprocessing step is necessary due to significant variations in the scales of different columns (Recency, Frequency, Monetary).

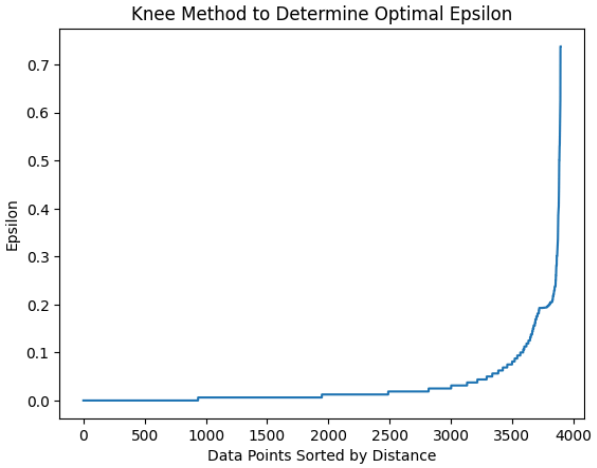
Determining the Optimal Epsilon

To effectively apply DBSCAN, selecting an appropriate value for epsilon (ε) is essential. Epsilon determines the maximum distance between two points for them to be considered as part of the same cluster. To identify the optimal value for ε, we employ the Nearest Neighbors approach, which involves:

1. **Calculating Distances**: Compute the distance to the second nearest neighbour for each point in the dataset. This is done because DBSCAN relies on a density-based clustering approach where the density threshold is defined by ε.
2. **Sorting Distances**: The computed distances are then sorted. This sorted list helps in visually identifying the point of maximum curvature, commonly known as the 'knee' or 'elbow', which suggests a good balance between including points within a cluster and excluding noise.
3. **Plotting for Visual Inspection**: The sorted distances are plotted to visually determine the knee of the curve. The optimal value for ε is typically chosen at this knee, as it represents a point where the rate of increase in distance indicates a transition to a less dense area.

choosing minPts: A common heuristic is to set minPts to twice the number of dimensions in the dataset. Since we have three dimensions (Recency, Frequency, Monetary), we'll start with minPts = 6.

With the data standardised and the optimal epsilon determined(around 0.25), we can proceed to build and test the DBSCAN model, aiming for effective customer segmentation



Let's fit the DBSCAN model and review the resulting clusters:

```
dbscan_estimated = DBSCAN(eps=0.25, min_samples=6)
clusters_estimated = dbscan_estimated.fit_predict(rfm_scaled)

# Add the estimated cluster labels to the dataframe
rfm_results['Estimated_Cluster'] = clusters_estimated

# Checking the distribution of estimated clusters and some descriptive statistics for each cluster
estimated_cluster_distribution = rfm_results['Estimated_Cluster'].value_counts()
estimated_cluster_stats = rfm_results.groupby('Estimated_Cluster')[['Recency', 'Frequency', 'Monetary']].mean()

print(estimated_cluster_distribution)
print(estimated_cluster_stats)
```

With the DBSCAN model using ε = 0.25 and minPts = 6, we've identified several distinct clusters, including a set of outliers:

Cluster Distribution and Characteristics:

- **Cluster 0 to Cluster 9**: Represents various segments of customers, each differing by their Recency, Frequency, and Monetary values.
- **Outliers (-1)**: Includes 132 customers who do not fit well into any of the defined clusters, likely due to having unique behaviours or being noisy in the dataset.

Key Cluster Insights:

- **Clusters 0-9** show a variety of purchasing patterns:
    - **Cluster 0**: 543 customers, recent moderate purchasers.
    - **Cluster 1**: 757 customers, slightly less recent and average spenders.
    - **Cluster 2**: 57 customers, recent high-frequency and high-monetary spenders.
    - **Cluster 3**: The largest, with 805 customers, less recent, lower frequency and spend.
    - **Cluster 4**: 342 customers, least recent, minimal spend.
    - **Cluster 5 and 7**: Similar profiles with moderate recency and higher spend.
    - **Cluster 6**: 675 customers, infrequent and low spend, not very recent.
    - **Cluster 8 and 9**: Smallest clusters, very high spenders and frequent buyers, very recent (26,6 respectively).

- **Outliers**: More diverse behaviours, potentially very high or very low values in any of the dimensions.

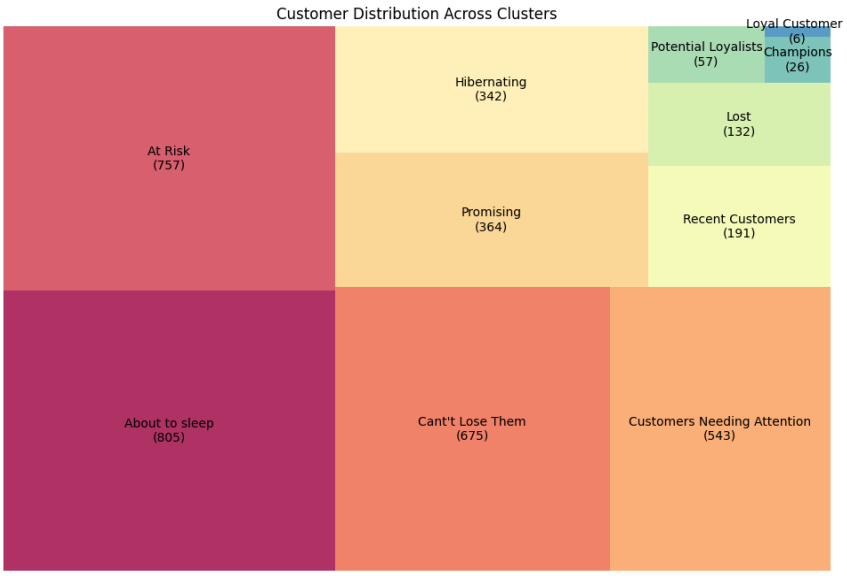
Review of Results: Strategic Business Value of Defined Customer Segments

Identification and Business Value of Customer Segments

1. **Champions (Cluster 8)**
    - **Characteristics**: These are the most engaged customers, with the highest frequency and monetary values, and very recent interactions.
    - **Business Value**: Being highly active and valuable, retention strategies such as exclusive offers, early access to new products, and loyalty programs are crucial to keep them engaged and maximise their lifetime value.
2. **Loyal Customers (Cluster 9)**
    - **Characteristics**: High spenders with a high frequency of purchases, moderately recent engagement.
    - **Business Value**: Encouraging continued engagement through personalised communications and loyalty rewards will solidify their loyalty and encourage even higher spending.
3. **Potential Loyalist (Cluster 2)**
    - **Characteristics**: Recent customers with relatively high frequency and monetary values.
    - **Business Value**: These customers show potential to become loyal customers. Engagement strategies like product recommendations and membership offers can help elevate their status.
4. **Recent Customers (Cluster 7)**
    - **Characteristics**: Customers who have recently made purchases with moderately high frequency and spending.
    - **Business Value**: These customers are at a critical juncture where timely and relevant engagement can convert them into more committed, loyal customers.
5. **Promising (Cluster 5)**
    - **Characteristics**: Recently engaged customers with higher than average frequency and spending.
    - **Business Value**: These customers could potentially shift into higher value segments. Tailored promotions and special offers based on their purchase history can enhance their engagement.
6. **Customers Needing Attention (Cluster 0)**
    - **Characteristics**: Moderately recent purchasers with average frequency and spending.
    - **Business Value**: Targeted communication strategies like email marketing or social media engagement can reinvigorate their interest and prevent migration to lower engagement clusters.
7. **About To Sleep (Cluster 3)**
    - **Characteristics**: Infrequent purchases and low spending, less recent engagement.
    - **Business Value**: Wake-up calls in the form of special, limited-time offers or feedback surveys can help reconnect these customers with the brand.
8. **At Risk (Cluster 1)**
    - **Characteristics**: The least recent visitors with minimal engagement and slightly high spend.
    - **Business Value**: High-priority recovery strategies are necessary, including win-back offers and personalised outreach to understand their disengagement reasons.
9. **Can’t Lose Them (Cluster 6)**
    - **Characteristics**: Customers who used to shop frequently but have shown a significant drop in recent activities.
    - **Business Value**: Aggressive re-engagement strategies like loyalty rewards, special discounts, or renewal benefits are needed to remind them of the brand's value.
10. **Hibernating (Cluster 4)**
    - **Characteristics**: Customers with older engagement, low frequency, and modest spending.
    - **Business Value**: Occasional check-ins and reactivation offers could stimulate interest and possibly revive their purchasing habits.
11. **Lost (Outliers -1)**
    - **Characteristics**: Customers with erratic and unpredictable purchasing patterns.
    - **Business Value**: Due to their unpredictable nature, these customers might require individual analysis to determine if efforts to reclaim them are cost-effective or feasible.

Visualisation of Customer Segmentation

The treemap presented here illustrates the distribution of various customer segments derived from the DBSCAN clustering process. Each segment is represented proportionally, offering a clear visual breakdown of our customer base



Data Mart Design: Dimensions and Metrics for Marketing Analysis

To support the marketing department's analysis needs effectively, it's essential to design a data mart that encapsulates key customer dimensions and metrics. This design will facilitate targeted marketing strategies, customer behaviour analysis, and performance monitoring.

Identification of Dimensions

1. Member
2. Geography
3. Time
4. Product
5. Store

Justification of Selected Dimensions

1. **Member**: It can provide significant benefits, particularly in enhancing customer-centric analyses and facilitating personalised marketing efforts.It could enrich our marketing strategies and decision-making by enhancing personalization,segmenting and targeting etc…
2. **Geography**: Important for regional analysis and localization of marketing efforts. Geography helps in understanding regional market penetration and tailoring promotions to fit local preferences and buying habits.
3. **Time**: Essential for trend analysis and tracking changes in customer behaviour over different periods. It enables tracking of seasonal trends, campaign effectiveness over time, and evolution of customer preferences.
4. **Product**: Allows analysis of sales performance at the product level, which is vital for inventory management, product development, and promotion strategies. It helps identify which products are successful within different customer segments or regions.
5. **Store:** Adding the "Store" dimension allows for a more detailed analysis of performance metrics at specific store locations and it enables the marketing department to customise campaigns based on the performance and consumer demographics of specific stores.

Identification of Measures

1. Total Sales Revenue
2. Customer Lifetime Value (CLTV)
3. Customer Acquisition Cost (CAC)
4. Recency
5. Frequency
6. Monetary

Justification of Selected Measures

1. **Total Sales Revenue**: A primary performance indicator that helps measure the direct financial outcome of marketing activities. It reflects the overall effectiveness of marketing strategies and promotions.
2. **Customer Lifetime Value (CLTV)**: A critical measure for evaluating the long-term value of customers segmented into various groups. CLTV helps in prioritising marketing efforts and determining the profitability of maintaining long-term relationships with different customer segments.
3. **Customer Acquisition Cost (CAC)**: Essential for evaluating the cost-effectiveness of marketing strategies aimed at acquiring new customers. It helps in understanding the investment required to attract a new customer and balancing it against the expected lifetime value of the customer.
4. **Recency**: Recency is crucial for identifying customers who have recently engaged with the brand, helping to prioritise them for retention strategies. It's a strong predictor of a customer's likelihood to respond to new offers, with more recent customers often more receptive to promotions.
5. **Frequency**: Frequency helps identify the most engaged and loyal customers. High-frequency customers are often prime targets for loyalty programs and upselling opportunities because their repeated interactions suggest a higher level of satisfaction and engagement with the brand.
6. **Monetary**: Monetary value is a direct indicator of a customer's economic value to the company. It helps in segmenting customers based on their spending levels, enabling more focused marketing strategies such as exclusive offers for high spenders or budget-driven promotions for lower spenders.

Adding "Cluster" as a measure in our data mart can be beneficial, but it's important to note that "Cluster" would technically be better classified as a dimension rather than a measure. Here's why and how it could be integrated:

Benefits:

- **Targeted Marketing**: Enables strategies tailored to the behaviours and needs of different customer groups identified by clusters.
- **Enhanced Personalization**: Supports more personalised marketing and product recommendations tailored to the specific preferences of each cluster.
- **Performance Analysis**: Facilitates comparison of key metrics across clusters to identify high-performing segments and areas for improvement.

Implementation Considerations:

- **Dynamic Updates**: Cluster assignments should be regularly updated to reflect changes in customer behaviour.
- **Integration**: Combine cluster data with other dimensions like Time, Geography, and Product to enable comprehensive multi-dimensional analysis.

Conclusion: This project successfully applied Data preparation,EDA,RFM segmentation and DBSCAN clustering to analyse and categorise customer data, enhancing our understanding of different purchasing behaviours. These insights will enable targeted marketing strategies, improving customer engagement and retention, and ultimately boosting the grocery store chain's profitability and customer loyalty.