RFM ANALYSIS

INTRODUCTION

This model is the result of a meticulous Python-based endeavor that harnessed the power of data. Our dataset, comprising three columns - member_number, date, and itemDescription, served as the bedrock for a comprehensive RFM analysis. The fusion of technology and data science has birthed a tool that promises to unravel the intricate web of customer behavior, thus empowering businesses to make informed decisions and optimize their strategies.

At its core, RFM analysis is a robust and data-driven methodology that allows us to gain profound insights into customer behavior. It segments customers based on three key dimensions:

1. Recency (R):This dimension examines how recently a customer made a purchase. It helps us identify whether a customer is a recent buyer or one who hasn't engaged with the business for a while. In the age of rapidly changing market dynamics, understanding the recency of customer interactions is crucial for tailored engagement strategies.

2. Frequency (F):Frequency represents the number of purchases a customer has made over a defined period. It sheds light on customer loyalty and engagement. High-frequency customers are the backbone of many successful businesses, as they generate consistent revenue.

3. Monetary Value (M):Monetary value, often referred to as the "Monetary" dimension, quantifies how much a customer has spent during their interactions with the business. This dimension helps us identify high-value customers, whose contributions significantly impact the bottom line.

By applying this methodology to our dataset, we embark on a quest to decipher the hidden patterns and trends that underlie customer behavior.
The journey of this model is guided by the principles of RFM analysis. The member_number, date, and itemDescription columns represent the crux of our data, forming a dynamic trifecta that offers insight into customer identity, purchase history, and product preferences.

In a world where data reigns supreme, businesses are in a constant pursuit of strategies that can help them not only retain their customer base but also nurture growth. RFM analysis, backed by the computational prowess of Python and Machine Learning, stands as a beacon of hope for enterprises striving to gain a competitive edge in the ever-evolving marketplace.

DATA UNDERSTANDING

I embarked on our data exploration journey by delving into the dataset, utilizing a range of essential functions to uncover its structure and concealed intricacies. The snippet below encapsulates our initial steps:

```
[ ]  1 df.describe()
```

	Member_number
count	38765.000000
mean	3003.641868
std	1153.611031
min	1000.000000
25%	2002.000000
50%	3005.000000
75%	4007.000000
max	5000.000000

```
[ ]  1 df.info()

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

```
[ ]  1 df.shape

(38765, 3)
```

```
1 df.isna().sum()

Member_number      0
Date               0
itemDescription    0
dtype: int64
```

The initial exploration revealed that the dataset comprises 38,765 rows and 3 columns, providing a fundamental understanding of its scale and dimensions. One notable highlight of this dataset is the absence of missing values, a vital characteristic that contributes to the creation of a smoother and more robust model.
Furthermore, I've scrutinized essential descriptive statistics such as mean, standard deviation, and median, which offer valuable insights into the dataset's central tendencies and variability. These statistics serve as a compass, guiding us in the process of understanding the data distribution and potential patterns.

Additionally, the data exploration endeavors extended to investigating the data types employed within the dataset. This aspect is pivotal in ensuring that we have a comprehensive grasp of the nature of our variables, which will be pivotal in subsequent stages of analysis and modeling.

As we navigate through this dataset, our commitment to thorough exploration and analysis remains unwavering, paving the way for the development of robust and insightful models that can unlock the dataset's full potential.
Here, in the realm of marketing strategy, the quest for insights led to a pivotal moment. I recognized the need to convert the data type of the 'date' column from 'object' to 'datetime.' This transformation enhances our ability to harness the temporal aspect of the data, enabling us to seize opportunities and craft strategies with precise timing.
Following the initial exploration, I turned our focus to addressing duplicates within the dataset. Given the nature of grocery data – where an entry is considered a duplicate only if all three values (member_number, date, and itemDescription) are identical – I meticulously identified and resolved these duplications. The code snippet below illustrates our process for detecting and managing these duplicates.

Lets work on duplicates, for this problem statement, we only consider it a duplicate if all three columns have same value

```
duplicates = df[df.duplicated(subset=['Member_number', 'Date', 'itemDescription'], keep=False)]
sorted_duplicates = duplicates.sort_values(by=list(duplicates.columns))
print(sorted_duplicates)
```

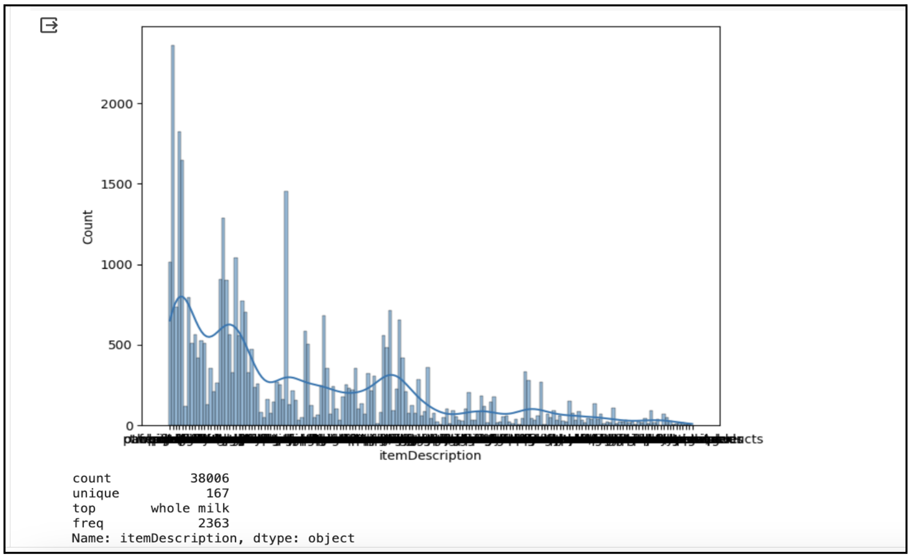
```
       Member_number       Date itemDescription
33098           1003 2014-02-27      rolls/buns
37649           1003 2014-02-27      rolls/buns
15099           1005 2014-09-01      rolls/buns
31248           1005 2014-09-01      rolls/buns
7532            1006 2015-06-14     frankfurter
...              ...        ...             ...
24043           4981 2015-10-01       margarine
8109            4988 2015-10-29      rolls/buns
24258           4988 2015-10-29      rolls/buns
33585           4992 2014-02-24       margarine
38136           4992 2014-02-24       margarine

[1491 rows x 3 columns]
```

```
df = df.drop_duplicates(subset=['Member_number', 'Date', 'itemDescription'])
```

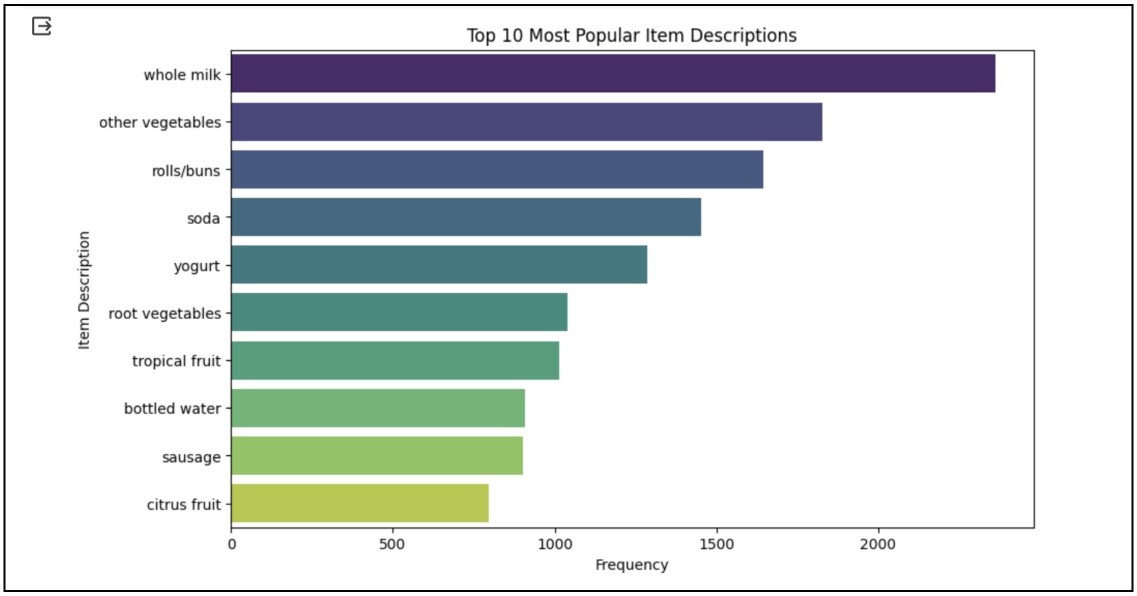
I transitioned to the data visualization phase, leveraging the power of visual representation to gain deeper insights. A series of histograms were meticulously crafted for each dataset column. These visual aids provided a comprehensive view, highlighting key statistics such as count, unique values, top occurrences, and frequencies.

This approach enabled us to extract valuable insights, answering questions like the most popular items among consumers and the characteristics of top and average buyers, ultimately enhancing our understanding of the dataset's nuances.

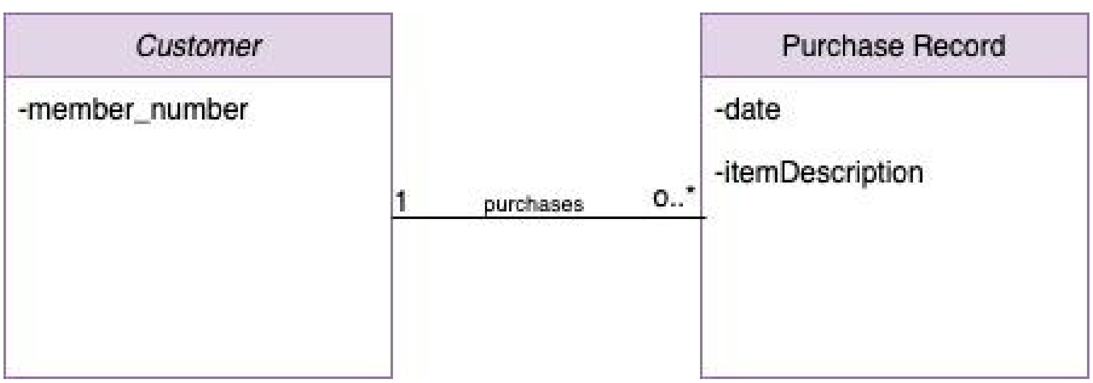


To further enhance the understanding of the dataset, I employed data visualization techniques and created a bar plot. This visual representation helped me delve into the monthly purchase trends, providing valuable insights into the dynamics of consumer behavior over time.

Continuing the data visualization journey, we constructed another insightful bar plot. This time, the focus was on revealing the top 10 items that dominated the purchase list, offering a clear and concise view of the most popular choices among consumers.



UML CLASS DIAGRAM



Class Identification:
1. Customer (Member):
   - Attributes: member_number

2. PurchaseRecord:
   - Attributes: date, itemDescription

Association Identification:
- Customer (Member) -[purchases]- PurchaseRecord

Cardinality Constraints:
- Customer (Member) -[1]--purchases [0..*]- PurchaseRecord

- The "Customer (Member)" class represents individual customers and has a one-to-many association with "PurchaseRecord." This means that each customer can have multiple purchase records (0 or more) over time.
- The "PurchaseRecord" class holds information about the date and item description of each purchase.
- The cardinality constraints ([1] and [0..*]) indicate that each customer has at least one purchase record (i.e., they made at least one purchase) but can have multiple purchase records, depending on their transaction history.

RFM SEGMENTATION

In the realm of RFM segmentation, the journey commenced with a deep dive into the "Recency" dimension. The key to understanding this aspect was establishing a reference point to gauge how recently a buyer had made a purchase.
To achieve this, I conducted a strategic maneuver. I pinpointed the maximum date in the dataset where purchase data was available, setting the stage for the reference date. This reference date was ingeniously chosen as the 1st of the next month. This selection, while convenient, serves as an effective benchmark for calculating the number of days since the last purchase. It is this precision and attention to detail that lays the foundation for the RFM analysis that follows.

Recognizing the constraints imposed by our dataset, I adapted our approach to assess the "Frequency" dimension. Instead of relying on traditional transaction records, I utilized the member number column as a proxy for understanding purchase frequency. This pragmatic solution allowed us to gauge how often customers were engaging with our offerings, despite the data limitations, and paved the way for a comprehensive RFM analysis.

RFM SEGMENTATION

```
max(df['Date'])
```

```
Timestamp('2015-12-30 00:00:00')
```

```
day = '2016-01-01'
day = pd.to_datetime(day)
recency = df.groupby(["Member_number"]).agg({"Date": lambda x: ((day-x.max()).days)})
```

```
recency.shape
```

```
(3898, 1)
```

```
frequency = df['Member_number'].value_counts()
freq = pd.DataFrame(frequency)

```

```
freq.shape
```

```
(3898, 1)
```

```
type(freq)
```

```
pandas.core.frame.DataFrame
```

```
df
```

	Member_number	Date	itemDescription	Year	Month
0	1808	2015-07-21	tropical fruit	2015	7
1	2552	2015-05-01	whole milk	2015	5
2	2300	2015-09-19	pip fruit	2015	9
3	1187	2015-12-12	other vegetables	2015	12

In the pursuit of understanding the "Monetary" dimension, I encountered a data gap where a direct "total bill amount" column was missing. To bridge this gap, a creative solution was crafted. I first established a "quantity" column by consolidating rows based on member number and date, enabling us to calculate the spending for a specific date.
To maintain convenience and uniformity, I assigned a fixed price of one dollar to all items. This approach allowed me to approximate the monetary aspect of customer interactions and subsequently conduct a comprehensive RFM analysis, despite the absence of a dedicated "total bill amount" column in our dataset.

```
[ ] df['Quantity'] = df.groupby(['Member_number', 'Date'])['itemDescription'].transform('nunique')
```

```
[ ] df
```

	Member_number	Date	itemDescription	Quantity
0	1808	2015-07-21	tropical fruit	3
1	2552	2015-05-01	whole milk	3
2	2300	2015-09-19	pip fruit	3
3	1187	2015-12-12	other vegetables	3
4	3037	2015-01-02	whole milk	3
...	...	...	...	...
38760	4471	2014-08-10	sliced cheese	3
38761	2022	2014-02-23	candy	3
38762	1097	2014-04-16	cake bar	3
38763	1510	2014-03-12	fruit/vegetable juice	3
38764	1521	2014-12-26	cat food	3

38006 rows × 4 columns

```
agg_df = df.groupby(['Member_number', 'Date']).agg({
    'itemDescription': ', '.join,
    'Quantity': 'first'
}).reset_index()

agg_df.head(10)
```

	Member_number	Date	itemDescription	Quantity
0	1000	2014-06-24	whole milk, pastry, salty snack	3
1	1000	2015-03-15	sausage, whole milk, semi-finished bread, yogurt	4
2	1000	2015-05-27	soda, pickled vegetables	2

```
[ ] fixed_price = 1.0

# Calculate the monetary value by multiplying 'Quantity' by the fixed price
agg_df['Monetary_Value'] = agg_df['Quantity'] * fixed_price

agg_df.head(10)
```

	Member_number	Date	itemDescription	Quantity	Monetary_Value
0	1000	2014-06-24	whole milk, pastry, salty snack	3	3.0
1	1000	2015-03-15	sausage, whole milk, semi-finished bread, yogurt	4	4.0
2	1000	2015-05-27	soda, pickled vegetables	2	2.0
3	1000	2015-07-24	canned beer, misc. beverages	2	2.0
4	1000	2015-11-25	sausage, hygiene articles	2	2.0
5	1001	2014-07-02	sausage, whole milk, rolls/buns	3	3.0
6	1001	2014-12-12	whole milk, soda	2	2.0
7	1001	2015-01-20	frankfurter, soda, whipped/sour cream	3	3.0
8	1001	2015-02-05	frankfurter, curd	2	2.0
9	1001	2015-04-14	beef, white bread	2	2.0

```
[ ] monetary = agg_df.groupby(['Member_number'])[["Monetary_Value"]].sum()
```

```
monetary.shape
```

```
(3898, 1)
```

```
[ ] RFM = pd.concat([recency, freq, monetary],axis=1)
recency.columns=["Recency"]
freq.columns=["Frequency"]
monetary.columns=["Monetary"]
```

```
[ ] RFM
```

K MEANS

In alignment with our specific dataset and modeling goals, RFM analysis integrated with K-means clustering provides an invaluable means of deciphering customer behavior. This approach tailors customer segmentation to our dataset's unique characteristics, enabling personalized marketing strategies. Utilizing the Elbow Method, we diligently determine the optimal number of clusters for our data, striking the right balance between granularity and simplicity.

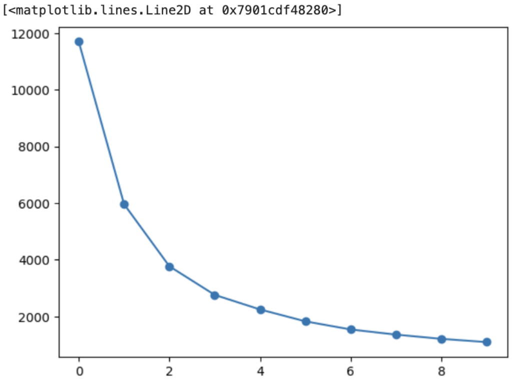
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled = scaler.fit_transform(RFM)
```

```
from sklearn.cluster import KMeans

inertia = []
for i in np.arange(1,11):
  kmeans = KMeans(n_clusters=i)
  kmeans.fit(scaled)
  inertia.append(kmeans.inertia_)

plt.plot(inertia,marker="o")
```

```
[<matplotlib.lines.Line2D at 0x7901cdf48280>]
```



This ensures that our segmentation aligns seamlessly with our business objectives, allowing us to craft strategies that resonate with distinct customer groups. This integration of RFM analysis, K-means clustering, and the Elbow Method empowers us to unlock the full potential of our data, facilitating data-driven marketing decisions that lead to enhanced customer engagement and business growth.

```
kmeans = KMeans(n_clusters=3)
kmeans.fit(scaled)
RFM["Clusters"] = (kmeans.labels_+1)
```

```
RFM
```

	Recency	Frequency	Monetary	Clusters
1000	37	13	13.0	1
1001	262	12	12.0	3
1002	124	8	8.0	3
1003	91	7	7.0	3
1004	323	21	21.0	1
...	...	...	...	...
4996	38	10	10.0	3
4997	5	6	6.0	3
4998	79	2	2.0	3
4999	6	16	16.0	1
5000	91	7	7.0	3

3898 rows × 4 columns

RESULTS

In our pursuit of meaningful insights, the results of this RFM analysis with K-means clustering have unveiled distinct customer segments.

Cluster 1, aptly named the "whales," stands out as the segment of high-value customers, consistently making substantial monetary contributions. Cluster 2 represents the "average" customers, demonstrating moderate spending behavior. Conversely, Cluster 3, identified as "lapsed customers," exhibits minimal engagement, with infrequent and minimal spending.

These clear distinctions among customer segments empower us to adopt tailored marketing strategies, maximizing our engagement with the high-value "whales," nurturing relationships with the "average" customers, and rekindling the interest of the "lapsed customers."

This strategic approach is poised to optimize our business's growth potential and customer satisfaction.

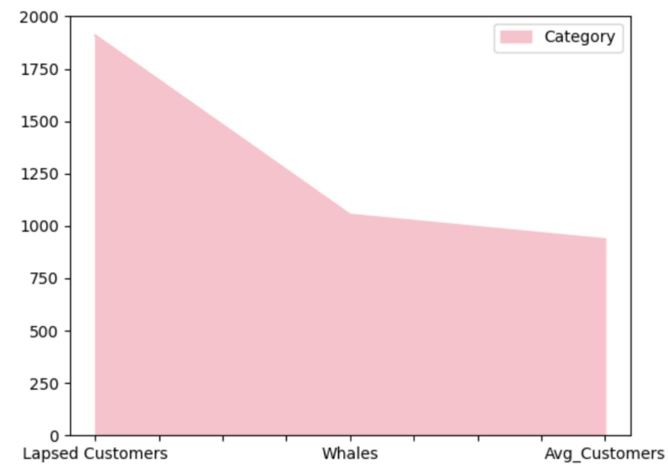
```
result = RFM["Category"].value_counts()
```

```
result
```

	Category
Lapsed Customers	1909
Whales	1053
Avg_Customers	936

```
result.plot(kind='area',color=["Pink"])
```

<Axes: >



DATA MART DESIGN

Dimensions:

1. Customer Dimension:
   - Attributes: Customer ID, Name, Contact Information, Demographics, Location
   - Justification: This dimension provides valuable insights into customer profiles, allowing for targeted marketing campaigns and personalized customer engagement.

2. Time Dimension:
   - Attributes: Date of Purchase, Month, Year, Season
   - Justification: The time dimension enables trend analysis, seasonality assessment, and optimized campaign timing, facilitating a better understanding of customer behavior over time.

3. Product Dimension:

- Attributes: Product ID, Item Description, Category, Price
- Justification: This dimension is essential for analyzing product popularity, optimizing inventory management, and formulating effective promotional strategies.

4. RFM Segment Dimension:
   - Attributes: RFM Segment (Whales, Average, Lapsed)
   - Justification: Segmenting customers based on Recency, Frequency, and Monetary attributes aligns with our analysis results and guides the creation of tailored marketing strategies.

Measures:

1. Total Sales:
   - Metric: Monetary Value
   - Justification: This key metric assesses revenue generation and customer spending trends, providing a core insight for marketing analysis.

2. Purchase Frequency:
   - Metric: Frequency of Purchases
   - Justification: Insights into customer engagement and loyalty, allowing for strategies that nurture and retain valuable customers.

3. Recency:
   - Metric: Days Since Last Purchase
   - Justification: This metric helps in evaluating customer retention and identifying lapsed customers, enabling re-engagement efforts.

4. Average Transaction Value:
   - Metric: Average Monetary Value per Transaction
   - Justification: Insights into individual purchase behavior and cross-selling opportunities, aiding in maximizing customer value.

5. Campaign Effectiveness:
   - Metrics: Conversion Rates, Response Rates, Return on Investment (ROI)
   - Justification: These metrics are pivotal for assessing the impact of marketing campaigns and optimizing strategies, ensuring the highest return on investment.

This data mart design is meticulously crafted to cater to the specific analysis needs of the marketing department, allowing for data-driven decision-making, improved strategy formulation, and enhanced customer engagement.
