Customer Management and Analytics

Hariharan Vels

Introduction

- An **Original Equipment Manufacturer (OEM)** produces components, parts, or products for another company's final product.
- Plays a crucial role in the supply chain by ensuring quality and cost efficiency.

Role of OEM in Supply Chain

- **Component Manufacturing** – Producing specialized parts for another company's product.
- **Assembly & Production** – Manufacturing and assembling products for branding companies.
- **Supply Chain Optimization** – Reducing costs and improving efficiency.
- **Aftermarket Support** – Providing spare parts and repair services.

Examples of OEMs in Different Industries

- **Automotive** – Bosch supplies parts to BMW, Ford, Toyota.
- **Electronics** – Samsung supplies display screens to Apple.
- **Computers & IT** – NVIDIA provides GPUs to ASUS, HP.
- **Aerospace** – Rolls-Royce supplies jet engines to Boeing, Airbus.

OEM vs. ODM

OEM	**ODM**
Produces parts/components	Designs and manufactures entire products
Example: Intel processors in HP laptops	Example: Generic smartphones rebranded by companies

Advantages of OEM in Supply Chain

- **Cost Efficiency** – Reduces production costs by outsourcing.
- **Focus on Core Business** – Companies can focus on branding, marketing.
- **High-Quality Components** – Specialization ensures top performance.
- **Faster Time to Market** – Companies can launch products quickly.

Customers in Supply Chain

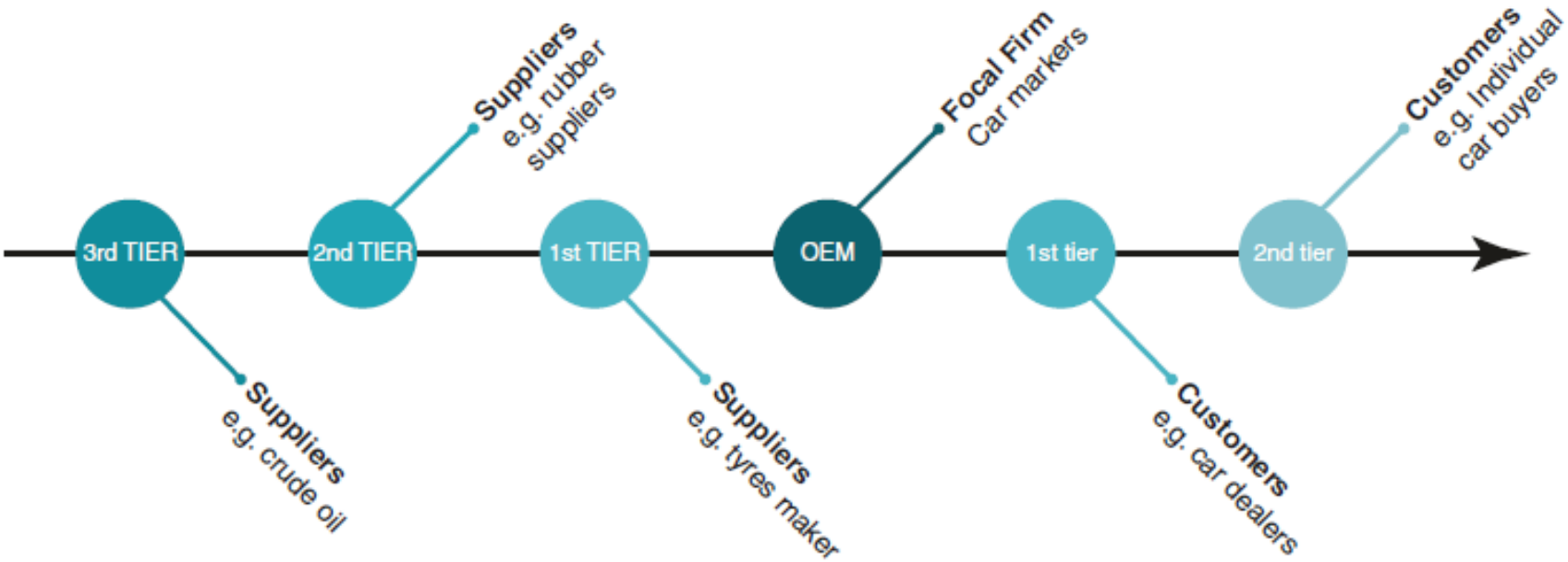

Figure: Customers in Supply Chain - A focal Firm's Perspective

Introduction

- Customers are the core of any supply chain.
- The ultimate goal of supply chain management is to meet and exceed customer expectations.
- Without customers, a supply chain has no purpose.

Why Customers Matter

- Successful businesses prioritize customer understanding.
- A well-designed supply chain maximizes value, efficiency, and responsiveness.
- Companies seek innovative partners to deliver the best products and services.
- Resilient supply chains ensure consistent and reliable service.

Market Research and Customer Insights

- Understanding customers offers many advantages.
- Companies invest in research to determine:
  - What products to offer
  - What value proposition to present
  - Which market segments to target

Case Study: Zara

- Zara thrives by focusing on customers.
- Strategic emphasis on:
  - Right products, at the right time, at the right price.
  - Analyzing sales data to adapt quickly to trends.
  - Using sentiment analytics to understand customer preferences.
- Regional tailoring of collections enhances customer satisfaction.

Customer Experience in E-Commerce

- Imagine ordering a webcam online but facing a long delivery wait.
- Customers prefer businesses with faster delivery times.
- A competitive market requires understanding and satisfying customer needs.

Customer-Centric Supply Chain

- A great customer experience requires a customer-focused approach.
- Emphasis on:
  - Identifying customer desires.
  - Strategically allocating resources.
  - Coordinating efforts to meet expectations.
- Every business, regardless of size or industry, should place customers at its core.

Benefits of having Customer Centric Supply Chain

Table: Benefit of understanding customers

Benefits	**Implications**
Better product design	Could potentially attract more customers, differentiate your business from rivals, etc.
Tailored products/individualization	Could also attract more customers, create great customer experience, and improve customer loyalty, etc.
Targeted pricing and promotion	More effective pricing and promotion strategies targeting specific customers at specific time and location.
Improved customer relations	Could enhance customer relationship management with deeper understanding of their needs and desires.
Better customer service	Likewise, could enhance customer service to better serve their needs, with fast response, personalized customer experience, etc.
Enhanced supply chain collaboration	Could build a customer-centric culture across the supply chain, improve overall supply chain efficiency and performance.
Better visibility and transparency	Could enhance information sharing and timely data exchange, thus improving overall supply chain performance.

Benefits of having Customer Centric Supply Chain - Contd

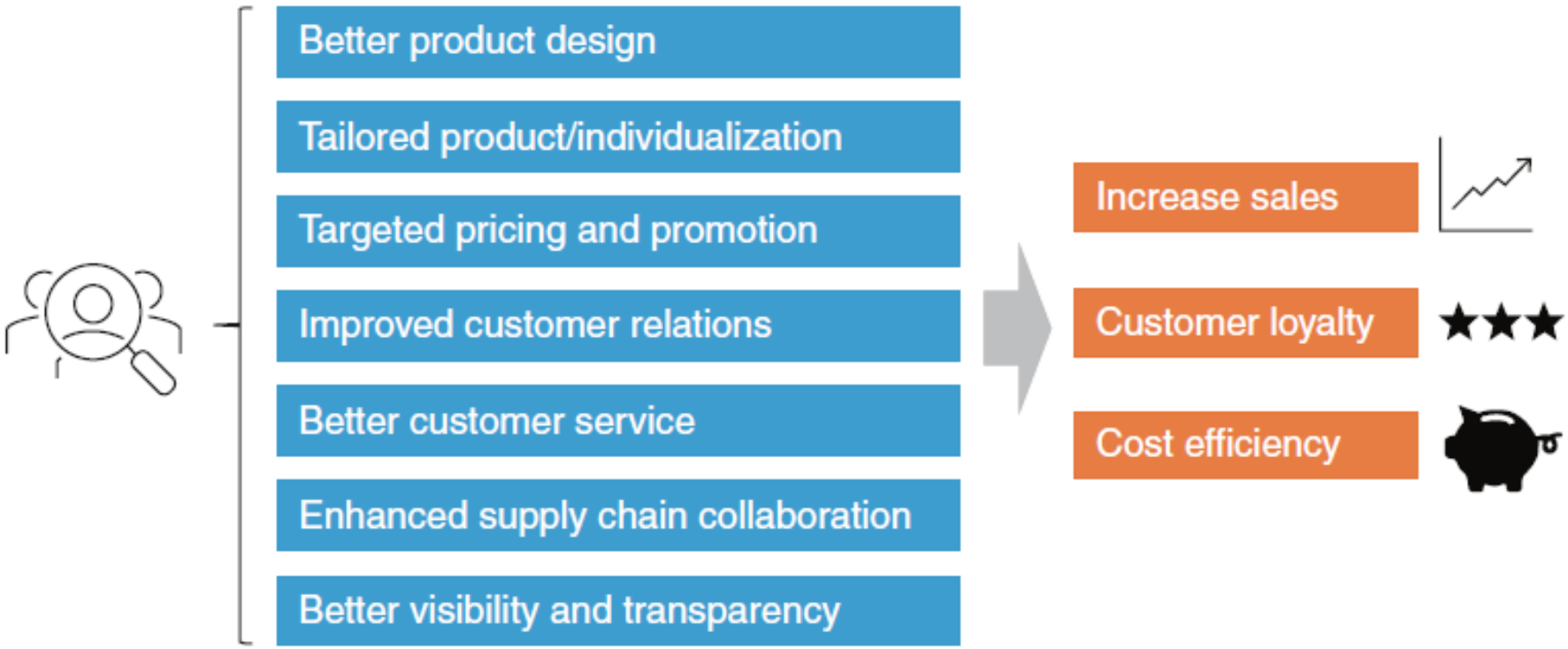

Figure: Benefits of having Customer Centric Supply Chain

Creating Customer Centric SC

- Developing a customer-centric supply chain is complex.
- Companies must realign their supply chains to meet customer needs.
- All members, from suppliers to customers, must collaborate.

Step One: Define Customers

- Identify who the real customers are in the supply chain.
- Focus on end customers rather than internal processes.
- Clear customer definition ensures alignment across the supply chain.

Step Two: Understand Customers' Real Needs

- Identify target market segments and value propositions.
- Use traditional and modern research methods:
  - Customer feedback, surveys, and focus groups.
  - Big data analytics and machine learning.
- Analyze purchase records, sales history, and demographic data.

Step Three: Translate Needs into Product Features

- Convert customer needs into tangible product or service features.
- Key product attributes:
  - Functionality, appearance, quality, reliability.
  - Availability, customer service, cost, speed, flexibility.

Step Four: Design Supply Chain Processes

- Align supply chain operations with customer expectations.
- Choose appropriate supply chain models:
  - Lean and agile supply chains for different needs.
- Select strong supply chain partners and build strategic relationships.

Step Five: Build Efficient Logistics Systems

- Consider outsourcing logistics to 3PL providers.
- Improve logistics through:
  - AI, IoT, and machine learning for route optimization.
  - Real-time tracking and fleet management.

Recap to 5 Steps of Building Process

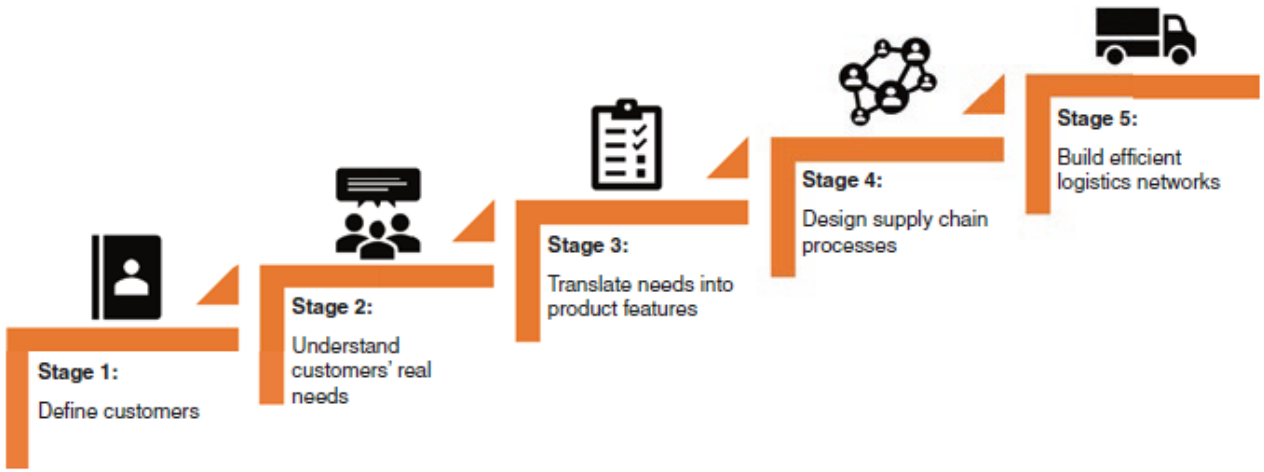

Figure: Recall for 5 Steps in Building Customer Based SC

Cohort Analysis

Cohort analysis has been used in social sciences for decades to study various fields like demography, psychology, epidemiology, and sociology (Glenn 2005). More recently, it has gained popularity in business analytics, particularly in marketing, to understand customer behavior by categorizing them into cohorts.

What Is a Cohort?

- A **cohort** is a group of individuals who share similar characteristics (e.g., gender, age, or culture) or experiences over a specified period (e.g., graduation year).
- In a business context, a cohort typically refers to customer groups based on when they first made a purchase.

What Is Cohort Analysis?

- **Cohort analysis** compares the behavior of different customer groups over time.
- It allows businesses to analyze customer retention and engagement rates.
- Organizations can gain valuable insights into customer focus areas.

Benefits of Cohort Analysis

- Develops specific pricing and promotional strategies.
- Helps measure the impact of product changes on customer acquisition and retention.
- Improves customer communication and relationships.

Illustration of Cohort Anlaysis

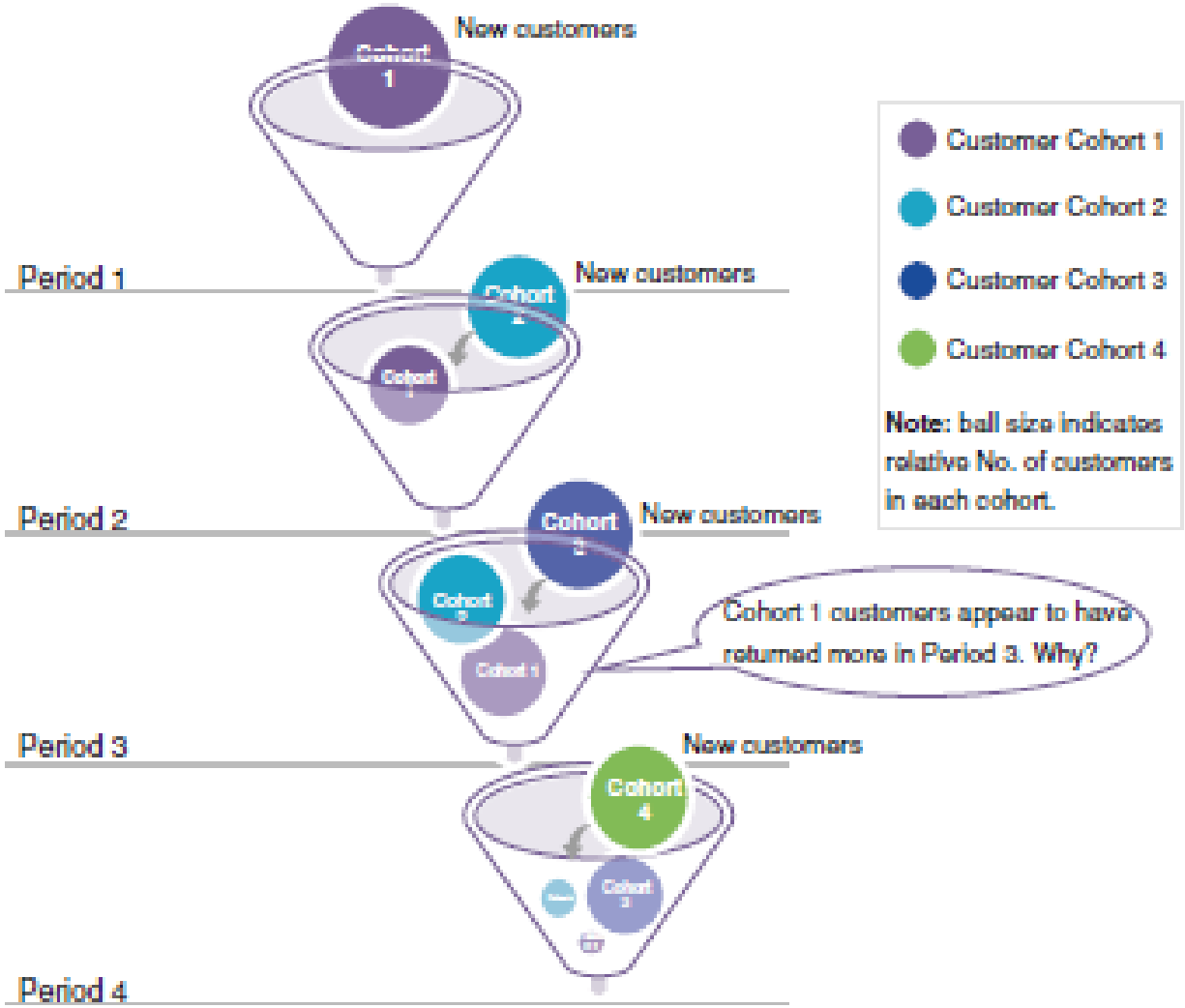

Figure: Illustration of Cohort Analysis

Example: Hair Salon Business

- A hair salon owner analyzes customer retention over a year.
- Examines the effectiveness of promotional activities.
- Uses cohort analysis to improve services and marketing strategies.

Steps for Cohort Analysis

- **Determine right cohort types** – Decide what types of cohorts to explore based on the analysis question (e.g., age, gender, or first purchase date).
- **Specify cohort size** – Define the range or intervals of data used to group customers (e.g., by week, month, or quarter).
- **Select metrics** – Choose values for comparison, such as retention rate or amount spent.
- **Set date range** – Define the period for analysis (e.g., comparing cohorts over the last 12 months).
- **Perform the cohort analysis** – Execute the analysis to obtain results.
- **Visualize and interpret results** – Use data visualization to represent findings and ensure interpretations align with the initial analysis questions.

Interpreting Analysis Results

After performing the analysis, we should attempt to interpret the results reasonably based on the initial questions that we want to answer in a specific business context.

Example: Cohort 1 Retention Rate

- For example, from the heatmap, we can see that cohort 1 (2010–12) appears to have the highest retention rate over the 12-month period compared to other cohorts.
  What is the reason for that?
- To answer the question, it would require certain business background information.
- You could build some initial propositions and validate them through further analysis.
- In our example, it turns out that the retail store was opened in December 2010, and most of the first cohort customers lived locally near the store.
- Therefore, they visited the store more often than other cohort customers. This cohort should be our focus.

Business Strategy

- As a business owner, you can make strategies to retain the cohort 1 customers.
- There are several spikes in retention rates during periods 2011–05, 2011–09, 2011–10, and 2011–11.
- Why did more customers return during these periods?
- Was the increased retention due to special events, promotional campaigns, or marketing efforts?
- After further analysis, we learned that these spikes were closely related to promotional campaigns and advertisements.

Christmas Shopping Impact

- The significant increase in retention in 2011–11 was also associated with Christmas shopping.
- Thus, the analysis and results enable us to learn whether our customer retention efforts are effective.
- The results help determine if we should continue the same campaigns at specific times.

Conclusion: Cohort Analysis

Overall, cohort analysis is helpful for business managers to understand customer loyalty.
While it cannot produce direct actionable results, it can provide useful insights into how to gain and retain customers.

Effective Strategies

By analyzing the results, you can think of effective strategies to increase customer retention by ascertaining what works and what does not, such as promotional campaigns, marketing efforts, and new product features.

Importing Required Libraries

The first step in the analysis is to import the necessary modules:

```
import pandas as pd
import numpy as np
import datetime as dt
import seaborn as sns; sns.set()
import matplotlib.pyplot as plt
```

Loading the Data

Next, we load the dataset using `pandas` `read_excel` function:

```
data = pd.read_excel('Retail_init_cleaned.xlsx')
```

We then check the structure and first few rows of the data:

```
data.info()
data.head()
```

The `describe()` function provides a summary of the data:

```
data.describe()
```

Identifying Outliers in Quantity

To detect any potential outliers in the `Quantity` column, we can use a boxplot:

```
data.Quantity.plot.box(vert=False)
```

To remove the outliers from the plot, we use the parameter `showfliers=False`:

```
data.Quantity.plot.box(vert=False, showfliers=False)
```

We can also count the number of records where the `Quantity` is greater than or equal to 28:

```
data[data.Quantity >= 28].count()
```

Analyzing Large Quantities

Next, we take a closer look at the data where `Quantity` exceeds 10,000:

```
Q10000 = data[data.Quantity > 10000]
Q10000
```

We can also analyze specific customer data. For example, for `CustomerID = 16446`, we extract their data as follows:

```
Customer16446 = data[data.CustomerID == 16446]
Customer16446
```

Similarly, for `CustomerID = 12346`:

```
Customer12346 = data[data.CustomerID == 12346]
Customer12346
```

Detecting Outliers in Unit Price

To detect outliers in the `UnitPrice` column, we use a boxplot again:

```
data.UnitPrice.plot.box(vert=False)
```

To remove the outliers from the plot:

```
data.UnitPrice.plot.box(vert=False, showfliers=False)
```

Most Expensive Products

To identify the products with the highest average `UnitPrice`, we group the data by `Description` and compute the mean:

```
data.groupby('Description')['UnitPrice'].mean().
nlargest()
```

Identifying High Quantity Sales

Lastly, we look for records where `Quantity` is greater than or equal to 8:

```
data[data.Quantity >= 8].count()
```

RFM Analysis

RFM analysis is a useful customer segmentation technique, widely adopted in marketing research. It helps categorize customers into various clusters, allowing businesses to target specific groups with tailored communications, engagement campaigns, and special treatments relevant to each group. In this section, we will introduce the basics of RFM analysis and use an example to demonstrate how it can be performed in Python.

What Is RFM?

RFM stands for Recency, Frequency, and Monetary. These metrics represent key customer values that help understand customer behavior:

- **Recency**: Measures how recent a customer has made a purchase. A high recency score suggests a customer is more likely to respond to your communications and engagements.
- **Frequency**: Measures how often a customer makes a purchase. A higher frequency score typically indicates that the customer is engaged and loyal.
- **Monetary**: Measures how much money a customer has spent in total or on average. A higher monetary score indicates that the customer is a big spender.

What Is RFM Analysis?

RFM analysis evaluates customers using a combination of these three metrics, which provides a more balanced view than relying on a single metric alone. For example, a tourist who makes a recent purchase might have a high recency score, but they may not be a customer who will return. RFM analysis answers critical business questions such as:

- Who are the most valuable customers?
- Who has the potential to become more valuable?
- Which customers can be retained?
- Which customers are more likely to respond to engagement campaigns?

Steps for RFM Analysis

There are several steps involved in performing an RFM analysis:

1. **Set the RFM analysis scope**: Determine the time period for analysis (e.g., 3, 6, or 12 months). This will set the reference point for calculating R, F, and M scores.
2. **Calculate R, F, M values**: Calculate the Recency (R), Frequency (F), and Monetary (M) values for each customer.
3. **Convert R, F, M values to scale scores**: Convert the raw R, F, and M values to a scale (typically 1–5). This step is important for creating a balanced comparison of customers.
4. **Apply RFM formula**: Aggregate R, F, and M scores into a single RFM score. A weighted average may be used depending on business priorities.
5. **Segment customers**: Use the RFM scores to segment customers into different categories (e.g., high-value, low-value).

Recency Calculation Example

To calculate the recency score, subtract a customer's most recent purchase date from today's date. The result gives the recency value (e.g., 1, 10, or 120 days).

$$\text{Recency (R)} = \text{Current Date} - \text{Last Purchase Date}$$

The smaller the value, the more recent the purchase, and the higher the recency score.

Frequency Calculation Example

To calculate frequency, either count the total number of purchases or calculate the average time between orders.

$$\text{Frequency (F)} = \frac{\text{Total Number of Purchases}}{\text{Time Period}}$$

For total purchases, the higher the value, the better. For the average time between orders, a smaller value is preferable.

Monetary Calculation Example

Monetary value can be calculated as the total money spent or average money spent per purchase:

$$\text{Monetary (M)} = \frac{\text{Total Money Spent}}{\text{Number of Purchases}}$$

Higher spending indicates that the customer is valuable, and they may require more attention.

Converting RFM Values to Scale Scores

Once the raw RFM values are calculated, the next step is to convert them into scale scores. A typical approach is to divide the values into five equal groups using quantiles.

$$\text{Score} = \text{Quantile Groups} \quad \text{(e.g., 1-5 scale)}$$

For Recency, a smaller value means a more recent purchase, so it gets a higher score. Similarly, the other metrics are scaled based on their distribution.

RFM Aggregation and Segmentation

Once the R, F, and M scores are assigned, you can either use these individual scores for segmentation (e.g., 1-1-1 for low-value customers, and 5-5-5 for high-value customers) or aggregate them into a single RFM score by averaging or summing them. Depending on your business, you may assign different weights to the R, F, and M metrics.

- Example: If monetary value is more important for your business, you can assign a higher weight to the M score.
- Once the aggregated scores are calculated, segment customers into different categories based on the final RFM score.

Business Use Cases for RFM Analysis

- **Customer Retention**: Identify high-value customers who are most likely to respond to engagement campaigns.
- **Targeted Marketing**: Focus on customers who have a high frequency of purchase but a low recency score.
- **Promotions and Discounts**: Offer special promotions to high monetary customers.
- **Predictive Analytics**: Use historical RFM data to predict customer lifetime value and future behavior.

Importing Modules

First, we import the necessary Python modules:

```
# Import modules
import pandas as pd
import numpy as np
import datetime as dt
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import seaborn as sns
```

These modules will allow us to handle data (pandas, numpy), work with dates (datetime), visualize the data (matplotlib, seaborn), and perform additional plotting.

Loading the Data

Next, we load the data from an Excel file into a pandas DataFrame:

```
# Load the data
df_uk = pd.read_excel('UK_Data.xlsx', index_col=0)
df_uk.head()
```

We specify the ‘index_col=0‘ to set the first column as the index of the DataFrame.

Getting the Reference Date

The reference date for calculating recency is the most recent 'InvoiceDate' in the dataset:

```
# Get the reference date
ref = df_uk.InvoiceDate.dt.date.max()
ref
```

This will give us the most recent date in the dataset.

Converting InvoiceDate to Date

We convert the ‘InvoiceDate‘ to a date-only format (without the time):

```
# Convert InvoiceDate to date only
df_uk.InvoiceDate = df_uk.InvoiceDate.dt.date
df_uk.head()
```

This step ensures that we have a clean date representation without the time component.

Calculating Recency, Frequency, and Monetary Values

Now, we calculate the three key metrics:

```
# Recency
recency = df_uk.groupby('CustomerID')['InvoiceDate'].
apply(lambda x: (ref - x.max()).days)
# Frequency
frequency = df_uk.groupby('CustomerID')['InvoiceNo'].
nunique()
# Monetary
monetary = df_uk.groupby('CustomerID')['Amount'].
sum()
```

- Recency: The number of days since the last purchase.
- Frequency: The total number of unique orders per customer.
- Monetary: The total amount spent by the customer.

Combining RFM Metrics

After calculating the individual metrics, we combine them into one DataFrame:

```
# Combine Recency and Frequency
df_rf = recency.to_frame(name='Recency').
join(frequency.to_frame(name='Frequency'))

# Add Monetary value
df_rfm = df_rf.join(monetary.to_frame(
name='Monetary'))
df_rfm.head()
```

This creates a DataFrame with the RFM scores for each customer.

Alternative RFM Calculation

An alternative way to create the RFM DataFrame is to use the ‘groupby‘ and ‘agg‘ functions:

```
# Alternative way to create RFM dataframe
RMF = df_uk.groupby(['CustomerID']).
agg({'InvoiceDate': lambda x: (ref - x.max()).days,
        'InvoiceNo': 'nunique',
        'Amount': 'sum'})

RMF.rename(columns={'InvoiceDate': 'Recency',
        'InvoiceNo': 'Frequency', 'Amount': 'Monetary'
        inplace=True)
RMF.head()
```

This approach combines the three metrics (Recency, Frequency, and Monetary) into a single DataFrame in one step.

Calculating Quantiles

Next, we calculate the quantiles for each RFM metric. These quantiles will help us assign scores to customers based on their RFM values:

```
# Get 4 quantiles values
Quantiles = df_rfm.quantile(q=[0.2, 0.4, 0.6, 0.8])
Quantiles
```

For example, we can get the 40th percentile for recency:

```
Quantiles.Recency[0.4]
```

Quantiles are used to categorize customers into different segments.

Defining Recency Score

Now, we define a function to assign a Recency score based on the quantiles:

```
# Recency score function
def Rscore(x, q, df):
        if x <= df[q][0.2]:
                return 5
        elif x <= df[q][0.4]:
                return 4
        elif x <= df[q][0.6]:
                return 3
        elif x <= df[q][0.8]:
                return 2
        else:
                return 1
```

This function assigns a higher score to more recent customers (lower recency value).

Applying Recency Score

We apply the ‘Rscore‘ function to the ‘Recency‘ column to generate scores:

```
# Create a copy and apply Rscore function
rfm_copy = df_rfm.copy()
rfm_copy['Rscore'] = rfm_copy['Recency'].
        apply(Rscore, args=('Recency', Quantiles))
rfm_copy.head()
```

This results in a new column ‘Rscore‘ in the DataFrame with scores for each customer based on their recency.

Defining Frequency and Monetary Scores

Similarly, we define a function to assign Frequency and Monetary scores based on their quantiles:

```
# Frequency and Monetary score function
def FMscore(x, q, df):
        if x <= df[q][0.2]:
                return 1
        elif x <= df[q][0.4]:
                return 2
        elif x <= df[q][0.6]:
                return 3
        elif x <= df[q][0.8]:
                return 4
        else:
                return 5
```

This function assigns scores to Frequency and Monetary values, with higher values receiving higher scores.

Exploratory Analysis

Functionality: Import necessary libraries for data manipulation, visualization, and handling dates.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt
import seaborn as sns
import os
%matplotlib inline
```

Loading the Dataset

Functionality: Load the Excel dataset into a Pandas DataFrame.

```
df = pd.read_excel("Online-Retail.xlsx")
```

Initial Data Screening

Functionality: Examine the dataset structure, check for missing values, and obtain summary statistics.

```
df.sample(n=30, random_state=42)  # View sample data
df.shape  # Get dimensions of dataset
df.info()  # Get dataset info
df.describe()  # Get summary statistics
df.isnull().sum()  # Count missing values
df.CustomerID.nunique()  # Unique customers
print('No.-of-Unique-StockCode:-', df.StockCode.nuniqu
print('No.-of-Unique-Descriptions:-', df.Description.n
```

Data Cleaning and Preparation

Functionality: Remove missing values and filter out negative transaction amounts.

```
df.dropna(subset=['CustomerID'], how='all', inplace=Tr
# Remove null CustomerIDs
df.shape
df['Amount'] = df.Quantity * df.UnitPrice  # Compute t
df = df[~(df.Amount < 0)]  # Remove negative transacti
df.shape
df.info()
df.to_excel('Retail_init_cleaned.xlsx', index=False)
# Save cleaned data
```

Preliminary Data Analysis

Functionality: Analyze dataset by ranking transactions and revenue by country.

```
>>> df.Country.value_counts()  # Count transactions per country
>>> df.groupby('Country')['Amount'].sum().sort_values(ascending=False)  # Revenue by country
>>> df.groupby('Country')['Amount'].sum().sort_values(ascending=False).plot('bar', figsize=(16,5))
# Plot revenue
```