Customer Segmentation Using KMeans Clustering with Python

By
Ahmad Faishal Akbar

Background & Objective

The marketing team from Online Retail Company in UK (United Kingdom) want to make a marketing strategy for next year (in this case: 2012). They need our help to perform **customer segmentation** based on customer behaviour such as **Recency** (number of days since the last transaction of the customer), **Frequency** (number of transactions in the last 12 months), and **Monetary Value** (total value that the customer has spent in the last 12 months).

Based on this situation, this project aims to perform **customer segmentation** based on **Recency**, **Frequency**, and **Monetary Value** using **KMeans Clustering** and give some **marketing strategy recommendations** for each customer segment.

About The Data

This is a transnational data set which contains all the **transactions** occurring between **01/12/2010** and **09/12/2011** for a UK-based and registered non-store online retail.The company **mainly sells unique all-occasion gifts**. Many **customers** of the company are **wholesalers**.

We got this data from Kaggle.com, if you'd like to see the data, please check in the link below:

https://www.kaggle.com/datasets/carrie1/ecommerce-data

About The Data

This dataset contains 8 variables as follows:

InvoiceNo : Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.

StockCode : Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.

Description : Product (item) name. Nominal.

Quantity : The quantities of each product (item) per transaction. Numeric.

InvoiceDate : Invoice date and time.

UnitPrice : Unit price. Numeric, Product price per unit in sterling.

CustomerID : Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.

Country : Country name. Nominal, the name of the country where each customer resides.

Methodology

- Data Cleaning & EDA
- Data Preprocessing
- Data Clustering
- Analyzing Clustering Result
- Conclusion & Recommendation

Data Cleaning & Exploratory Data Analysis (EDA)

In this step, We performed various Data Cleaning & EDA processes such as **removing duplicate data, validating numeric data, performing descriptive statistics**, and more. If you'd like to review these processes and the entire processes, please check my notebook here:

https://colab.research.google.com/drive/1uK4Q_sY4_ZLkmhtLyaCMgkDk0AvdpLth?usp=sharing

Methodology

- Data Cleaning & EDA
- Data Preprocessing
- Data Clustering
- Analyzing Clustering Result
- Conclusion & Recommendation

Data Preprocessing: Features Creation

Recency: the number of days since the last transaction of the customer - the lower it is, the better, since every company wants its customers to be recent and active. the **transactions data** occurred between **01/12/2010** and **09/12/2011**. Let's assume the analysis is performed 2 days after the last purchase. We take 2 days to avoid zeros in Recency. We make **11/12/2011** as the **current date**. We could get the Recency days by **subtracting current date with the last time purchased of the customer (max InvoiceDate for each customer)**.

Data Preprocessing: Features Creation

Frequency: the number of transactions in the last 12 months. We could get it by counting number of transactions (**counting unique InvoiceNo**) for each customer.

Monetary Value: the total values that the customer has spent with the company in the last 12 months. We could get it by **multiplying UnitPrice with Quantity** and get the **total sum for each customer**.

Notes: the 12 months is a standard way to do this, but it can be chosen arbitrarily depending on the business model and the lifecycle of the products and customers.

Data Preprocessing: Features Creation

Here it is the code to get them

```
#Getting monetary value
df_rfm['MonetaryValue'] = df_rfm['UnitPrice'] * df_rfm['Quantity']
df_rfm

#Getting current date
from datetime import datetime
current_date = datetime(2011,12,11)

#Calculating recency, frequency, and monetary value for each customer
rfm = df_rfm.groupby('CustomerID').agg({'InvoiceDate' : lambda x: (current_date - x.max()).days,
                                        'InvoiceNo' : lambda x: x.nunique(),
                                        'MonetaryValue' : 'sum'})
rfm.columns = ['Recency', 'Frequency', 'MonetaryValue']
```

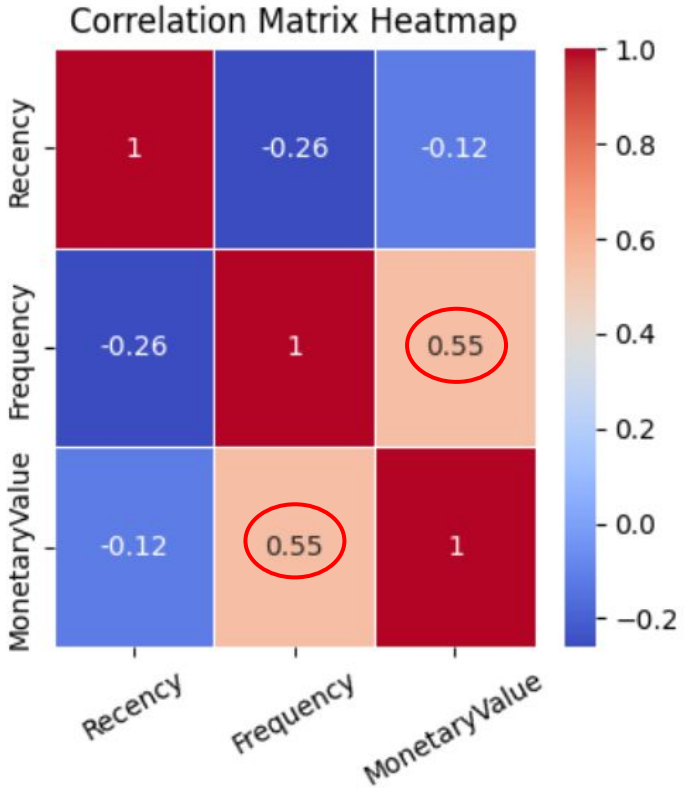
Data Preprocessing: Features Creation

CustomerID	Recency	Frequency	MonetaryValue
12346	326	1	77183.60
12347	3	7	4310.00
12348	76	4	1806.84
12349	19	1	1757.55
12350	311	1	334.40
...	...	...	...
18280	278	1	180.60
18281	181	1	80.82
18282	8	2	178.05
18283	4	16	2004.50
18287	43	3	1809.68

4339 rows × 3 columns

Data Preprocessing: Features Correlation

There is a **moderate correlation** between **Frequency** and **Monetary Value**. The correlation is expected, as customers tend to spend more when they make purchases more frequently.



Data Preprocessing: KMeans Assumptions

- The first assumption is that all **features** have **symmetrical distributions** (**normal distribution**).

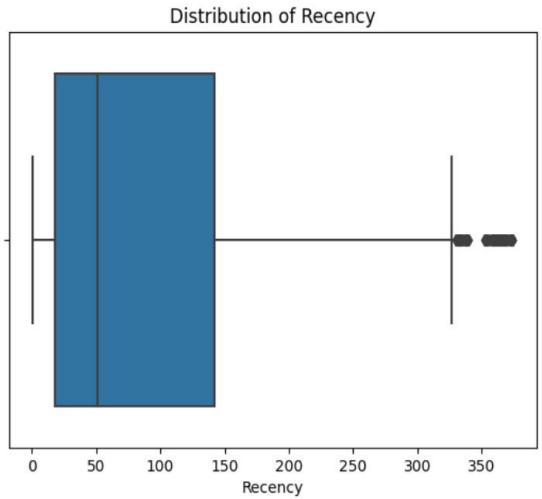
- The second assumption is that all **features** have the **same average values**.

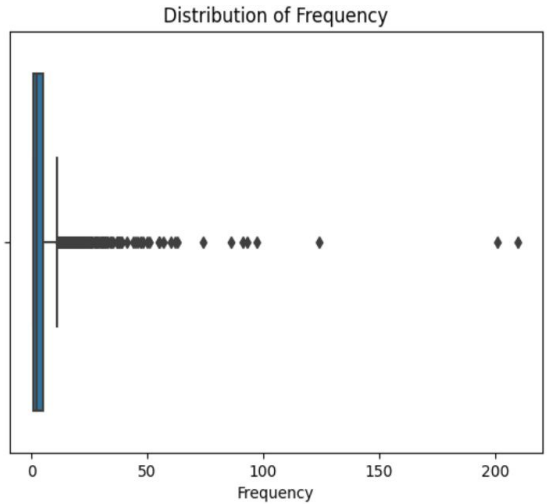
Data Preprocessing: Descriptive Statistics

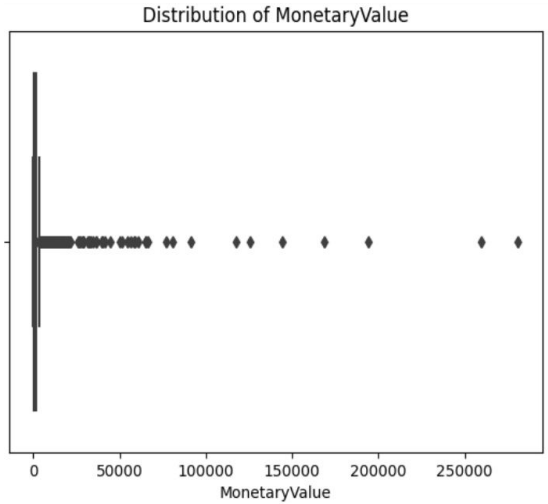
	Recency	Frequency	MonetaryValue
count	4339.000000	4339.000000	4339.000000
mean	93.041484	4.271952	2045.263983
std	100.007757	7.705493	8990.497308
min	1.000000	1.000000	3.750000
25%	18.000000	1.000000	305.955000
50%	51.000000	2.000000	664.000000
75%	142.500000	5.000000	1657.210000
max	374.000000	210.000000	280986.500000

All these features have **different scale and range of values as well as different value of mean and median**. **Recency** and **Monetary Value** have **very large standard deviation**. We need to see the distribution of each feature.

Data Preprocessing: Features Distribution

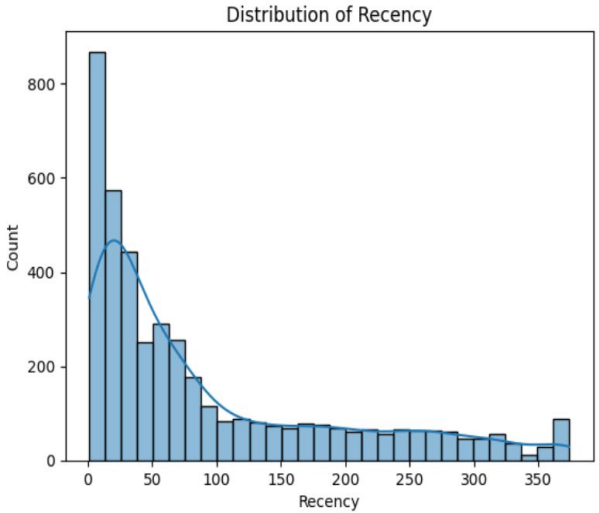


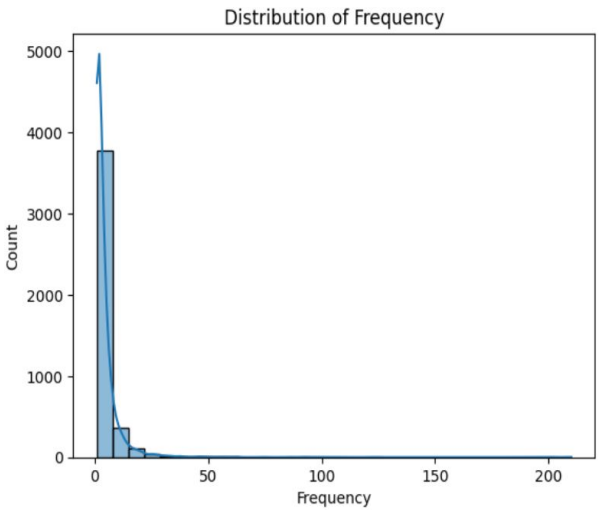


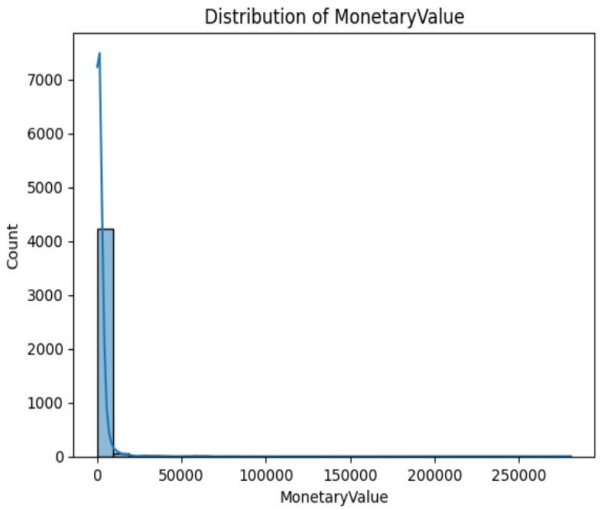


It seems a lot of outliers in Frequency and Monetary Value. But, **We can't treat them as outliers** considering many **customers** of the company are **wholesalers**. And, We can handle this by **scaling** the data.

Data Preprocessing: Features Distribution

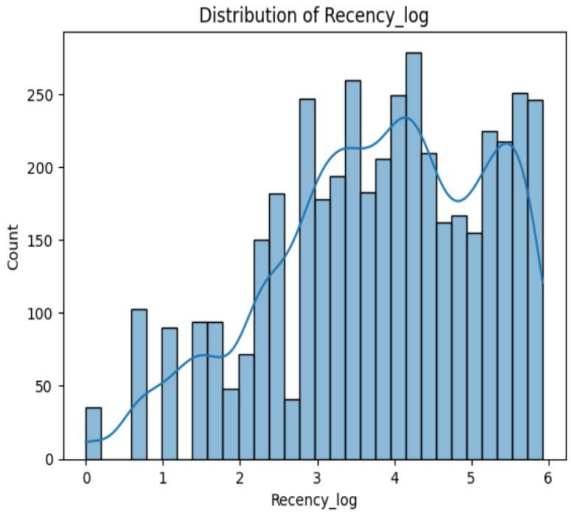


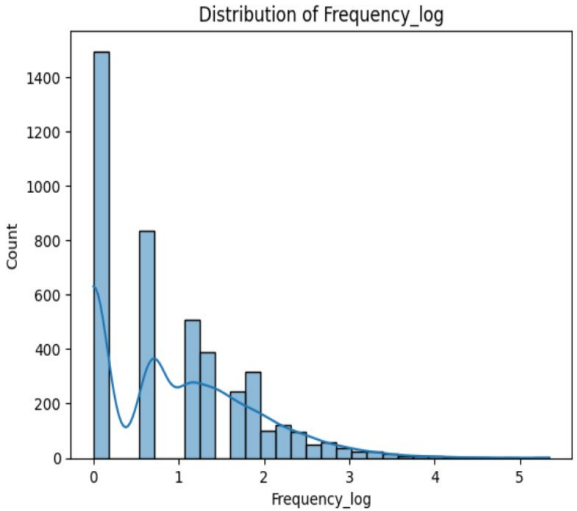


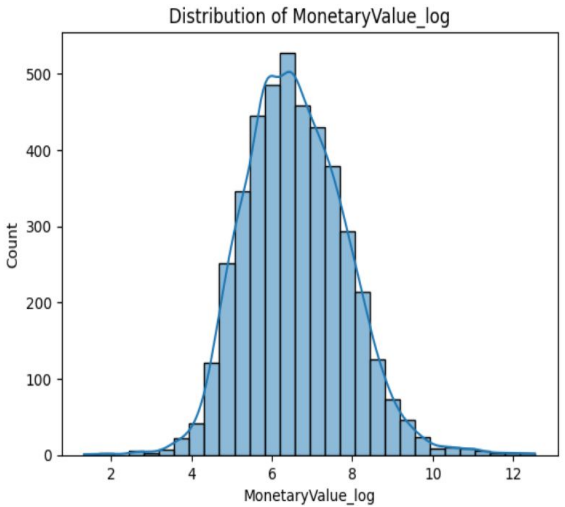


All the features have **right-skewed distribution**. Hence, We need to perform **data transformation** to meet KMeans first assumption.

Data Preprocessing: Data Transformation







We performed logarithmic transformation. The **features distribution are not skewed** now.

Data Preprocessing: Data Transformation

The features still have **different average values**. Therefore, We need to perform data scaling.

Here is the result after scaling the data:

	Recency_log	Frequency_log	MonetaryValue_log
count	4339.000000	4339.000000	4339.000000
mean	3.800803	0.944320	6.582713
std	1.383560	0.900861	1.262451

	Recency_scaled	Frequency_scaled	MonetaryValue_scaled
count	4339.000000	4339.000000	4339.000000
mean	-0.063328	0.156062	0.049975
std	0.668721	0.559736	0.747254

Now, **All the features have same average values** (around zero). So, We can perform KMeans Clustering.

Methodology

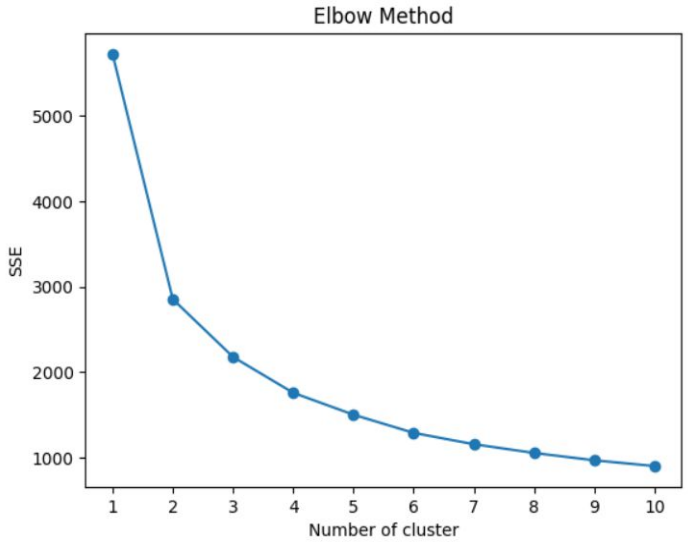
- Data Cleaning & EDA
- Data Preprocessing
- Data Clustering
- Analyzing Clustering Result
- Conclusion & Recommendation

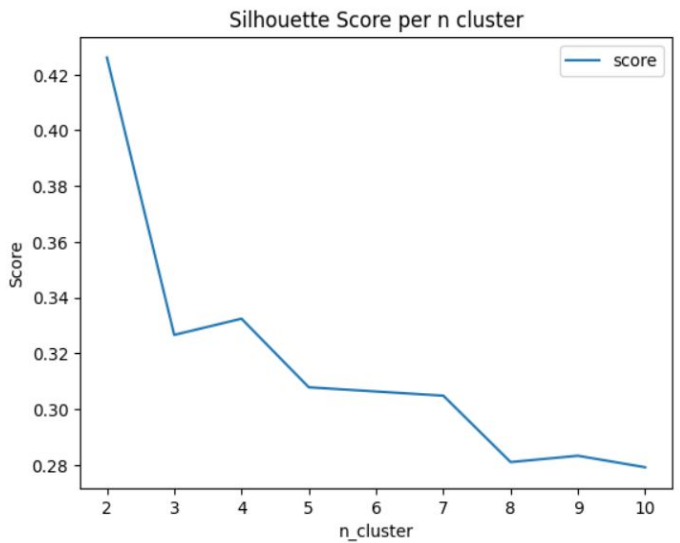
Data Clustering: Determining Number of Clusters

There are several ways to determine number of clusters such as **Elbow Method** (visually) and **Silhouette Analysis** (mathematically). We performed both of them.

- The **Elbow Method plots the sum of squared errors** (the sum of squared distances from each data point to their cluster center) for each number of segments. We then look at the chart to **identify where the decrease in SSE slows down and becomes somewhat marginal**. That point **looks like an elbow** of a bended arm and it **shows where there are diminishing returns by increasing the number of clusters**. This point **represents the optimal number of clusters** from a sum-of-squared errors perspective.
- The **silhouette score** is **a measure of how similar an object is to its own cluster** (cohesion) and **how different it is from other clusters** (separation). The silhouette score **ranges from -1 to +1**, where a **high value signifies that our object is well clustered**, and a **low value points out a poor clustering** choice.

Data Clustering: Determining Number of Clusters





Based on both of charts, 2 clusters have the highest silhouette score and the SSE starts decreasing slow down from that point. On the other hand, 4 clusters have the 2nd highest silhouette score and I think it will give us more insights than 2 clusters. Hence, **4** is the **optimal number of cluster.**

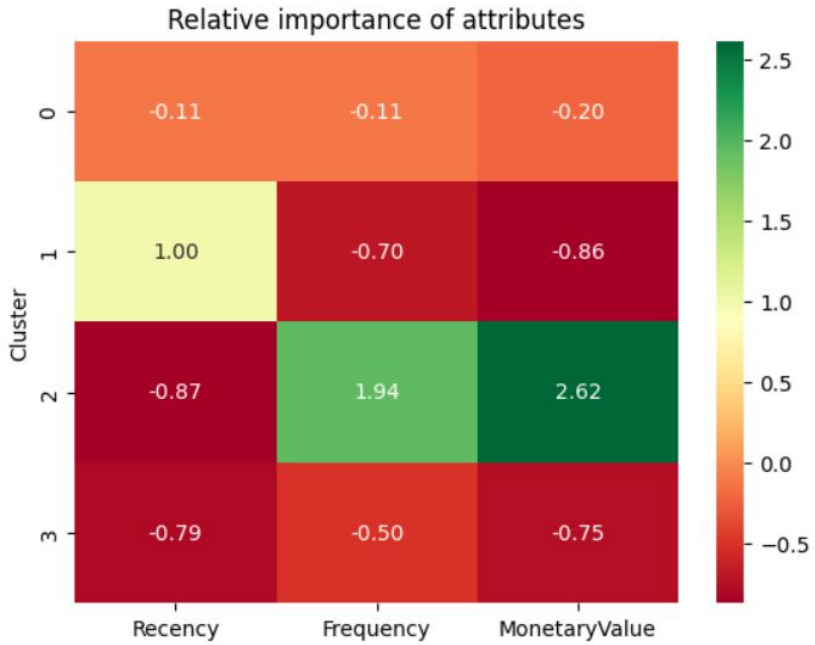
Data Clustering: KMeans Clustering

```
#data modelling
kmeans = KMeans(n_clusters=4, random_state=51)
kmeans.fit(rfm_scaled)
labels = kmeans.labels_

#assigning labels
rfm_labeled = rfm.assign(Cluster = labels)

rfm_labeled
```

	CustomerID	Recency	Frequency	MonetaryValue	Cluster
0	12346	326	1	77183.60	0
1	12347	3	7	4310.00	2
2	12348	76	4	1806.84	0
3	12349	19	1	1757.55	3
4	12350	311	1	334.40	1
...	...	...	...	...	...
4334	18280	278	1	180.60	1
4335	18281	181	1	80.82	1
4336	18282	8	2	178.05	3
4337	18283	4	16	2004.50	2
4338	18287	43	3	1809.68	0

4339 rows × 5 columns

Data Clustering: Relative Importance of Cluster Attributes



In general, **we want our segments to differ from the overall population, and have distinctive properties of their own.** We can use this technique to identify relative importance of each attribute. First, we calculate the average RFM values for each cluster. Then, we do the same for the total population. Finally, we divide the two, and subtract 1 from the result. Subtracting 1 ensures 0 is returned when cluster average equals population average. **The further that ratio is from zero, the more important that attribute is for defining a specific cluster compared to the population average.**

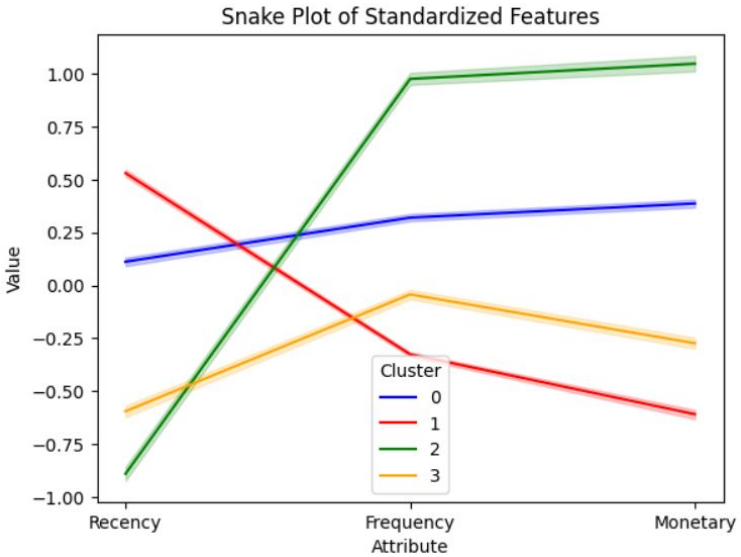
All attributes are **important enough** for **defining a specific cluster** compared to the population average.

Methodology

- Data Cleaning & EDA
- Data Preprocessing
- Data Clustering
- Analyzing Clustering Result
- Conclusion & Recommendation

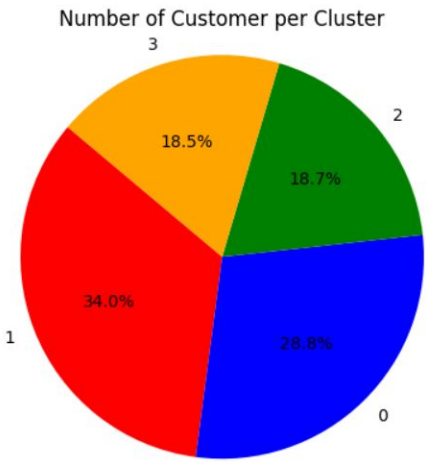
Analyzing Clustering Result

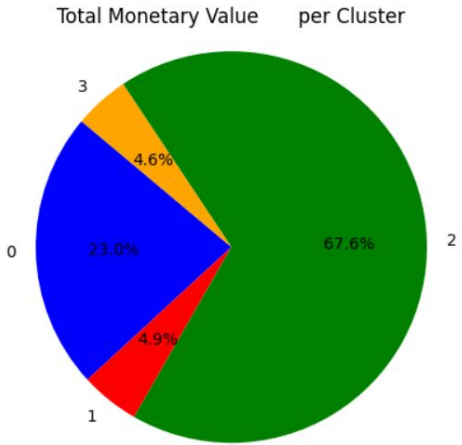
Cluster	Recency	Frequency	MonetaryValue
0	83.21	3.82	1628.20
1	185.78	1.27	293.21
2	12.30	12.57	7399.25
3	19.26	2.12	504.31

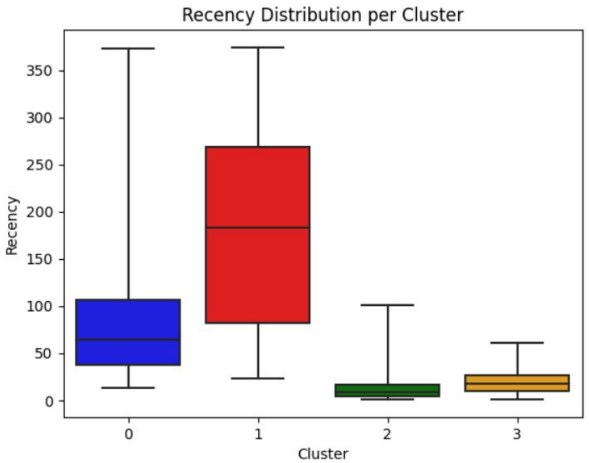


Based on **average values** for each cluster, **Cluster 2** is **the most valuable customers** that have **highest RFM values**. Otherwise, **Cluster 1** is **the most at risk customers** that have **lowest RFM values**. **Cluster 3** is **more recent than Cluster 0**, but it has **lower Frequency and Monetary Value than Cluster 0**

Analyzing Clustering Result







Cluster 1 has the **the highest number of customers (34%)**. But, **Cluster 1** along with **Cluster 3** have **the least impact on company's revenue (4.6% - 4.9%)**. **Cluster 3** is **more recent than Cluster 1**. **Cluster 2** along with **Cluster 3** have **the lowest number of customers (18.5% - 18.7%)**, but **Cluster 2** has **the most significant impact** on the company's **revenue (67.6%)**. **Cluster 0** has **moderate number of customers (28.8%) and impact** on the company's **revenue (23%)**, but it has **high range of recency days** and It is **less recent than Cluster 2 and Cluster 3.**

Methodology

- Data Cleaning & EDA
- Data Preprocessing
- Data Clustering
- Analyzing Clustering Result
- Conclusion & Recommendation

Conclusion & Recommendation

Based on the clustering results, we can outline the characteristic for each cluster and assign a name to each of them as follows.

Cluster	Description	Name
0	Customers who have **spent a good amount** but long ago (**not purchased recently**).	**Need attention**
1	Customers who have **not purchased recently** and/or tend to **spent less overall**.	**At risk**
2	Customers who have **purchased most recently**, **most frequently** and **spent the most**.	**Champions**
3	Customers who may have **purchased recently**, but they **do not tend to purchase frequently**.	**Recent**

Conclusion & Recommendation

Based on the characteristic of each segment (cluster), We make some marketing strategy recommendations as follows.

Segment	Recommendation
Champions	• Offer **exclusive rewards** or loyalty perks to maintain their frequent purchases and enhance their lifetime value. • Tailor **promotions or product recommendations** based on their past purchases to further entice them to buy. • Provide **premium customer service**, early access to new products, or dedicated support to reinforce their loyalty. • Suggest **complementary or higher-value products** to increase their average order value during their frequent purchases. • Encourage them to refer others by **offering incentives**, leveraging their satisfaction to attract new customers. • Seek **their opinions** to improve services/products, making them feel valued and involved in business decisions. • Maintain regular communication through **targeted campaigns**, ensuring they feel engaged and valued beyond transactions.

Conclusion & Recommendation

Based on the characteristic of each segment (cluster), We make some marketing strategy recommendations as follows.

Segment	Recommendation
Need attention	• Launch **targeted campaigns** offering **incentives or discounts** to encourage their return, reminding them of their past positive experiences. • **Recommend products** based on their previous purchases, showing new or updated items that might interest them. • Provide **exclusive offers** for their return, such as **limited-time promotions** or **loyalty rewards** for their next purchase. • Reach out through **various channels** (email, social media, or personalized messages) to maximize touchpoints and visibility. • Seek **feedback** about their past experiences and reasons for the lapse, showing interest in their needs and aiming to improve services. • Offer loyalty perks or **rewards** upon their return, re-establishing their value to the business.

Conclusion & Recommendation

Based on the characteristic of each segment (cluster), We make some marketing strategy recommendations as follows.

Segment	Recommendation
Recent	• Create **time-sensitive promotions or flash sales** to prompt immediate action and encourage impulse purchases. • **Package related items** together at a **discounted** rate, **incentivizing** multiple purchases at once. • Send **personalized reminders** or **notifications** about **abandoned carts** or products they previously showed interest in. • Use their **purchase history to suggest products** they might like, enhancing their shopping experience. • Ask for **feedback** on their experience and preferences, tailoring offerings to match their needs better. • Implement a loyalty program that rewards more purchases with exclusive perks or discounts, **encouraging repeat buying behavior.** • Use **multi-channel marketing** (email, social media, SMS) to keep them engaged, offering various touchpoints to increase visibility.

Conclusion & Recommendation

Based on the characteristic of each segment (cluster), We make some marketing strategy recommendations as follows.

Segment	Recommendation
At risk	• Initiate **targeted win-back campaigns** with attractive offers or discounts to encourage their return and reignite interest. • Tailor **promotions and product recommendations** based on their past purchases or browsing behavior, suggesting relevant and enticing deals. • Provide **incentives** like discounts, freebies, or loyalty points upon their return, demonstrating appreciation for their past patronage. • Send **surveys** to understand their disengagement reasons, enabling improvements tailored to their needs. • Utilize **targeted messaging** through **various channels** to reconnect and remind them of your brand.

References

datacamp.com:

- Cleaning Data in Python
- Exploratory Data Analysis in Python
- Customer Segmentation in Python

https://www.putler.com/rfm-analysis/

https://help.klaviyo.com/hc/en-us/articles/17797937793179

Thank You . . .