Project Report
Customer Segmentation of a Photo Company

Gokay Bulut

Note:
Since the company owns all intellectual property rights, this is not the original report provided to the company. Rather, it is a report explaining my work.

Summary

In this project, I studied customer behaviors of a Photo Company. First, I performed a cohort analysis and calculated customer retention rates through the year of 2016. Next, I analyzed RFMT (Recency, Frequency, Monetary Value and Tenure) values. Finally, I used K-Means Clustering, and employed Snake Plots, Relative Importance of Segment Attributes Heatmap and Silhouette Analysis to identify, verify and visualize the clusters. In conclusion, a 3-segment solution was recommended.

1. INTRODUCTION

1.a. General

All customers have diverse needs and desires and respond to marketing campaigns in different ways. Mass marketing strategies might improve the sales, but it would be a better approach to do the homework and analyze the customers based on their purchase behavior. As the business grows, segmenting customers can significantly improve marketing performance, making campaigns more relevant to more of the customers, ultimately increasing response rates and sales.

1.b. Problem

A Photo Company provided its data and wanted to gain more information on customer behavior. To boost the sales and excel in the service it provides, the company wants to target each segment separately in its marketing campaigns. My goal is to group customers into segments to understand high level trends better by providing insights on metrics across product/service and customer lifecycle.

1.c. Data Set

My dataset comes from the customer transactions dataset of the photo company. The data was obtained from the company. This dataset has 98,572 data points in total. Each record has the features below (translated to English):

- 'Inv_No' - number for each transaction (integer).
- 'Inv_Date' - time of the transaction (string).

- 'name’ - name of the customer (string) (dropped for privacy reasons).
- 'lastname' - lastname of the customer (string) (dropped for privacy reasons).
- 'Cust_ID' - unique number for identifying the customers (integer).
- 'Photo_Type' - types of photos taken (string).
- 'Amount' - amount paid by the customer for the transaction (float).
- 'Notes’ - notes on the transaction (string).
- reviewer_id - unique number of the reviewer

2. DATA WRANGLING & EXPLORATORY DATA ANALYSIS (EDA)

2.1. Initial Understanding

The initial look of the data set after translating column names and dropping the ‘name’ and ‘lastname’ columns for privacy reasons:

	Inv_No	Inv_Date	Cust_ID	Photo_Type	Amount	Notes
0	43891	01/01/2016	16106	amatör	29.78	Co k acele
1	43892	01/01/2016	10570	pasaport	32.94	Acele, bir an once yapilamli
2	43893	01/01/2016	13796	vesikalik.	29.45	Bizim
3	43894	01/01/2016	10246	Okul	23.94	Liste -- oncelikli
4	43895	01/01/2016	5158	pasaport	23.58	Tanidik

2.2. Information – info()

One of the basic and common ways to examine the data is using “info()” method. It is simple but tells a lot.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98572 entries, 0 to 98571
Data columns (total 8 columns):
Inv_No         98572 non-null int64
Inv_Date       98572 non-null object
name           98572 non-null object
lastname       98572 non-null object
Cust_ID        98572 non-null int64
Photo_Type     91173 non-null object
Amount         98572 non-null float64
Notes          93946 non-null object
dtypes: float64(1), int64(2), object(5)
memory usage: 6.0+ MB
```

- What to learn from this information:
  - The shape is 98572 observations (records or rows) and 8 columns (or variables).
  - There is redundancy in columns. Since ‘Cust_ID’ provides enough information on the customers, I dropped the 'name' and 'lastname' columns in order to take care of the redundancy and keep privacy of the customers who provided the reviews to the company.
  - Columns 'Photo_Type' and 'Notes' have missing values. The type of photos may be of interest.
  - I lowercased 'Photo_Type' column and stripped dots at the end of some entries.
  - I am not interested in notes on transactions for my analysis. So, I can drop it.
  - ‘Inv_Date' column is of type string. I will convert to datetime type for better analysis.

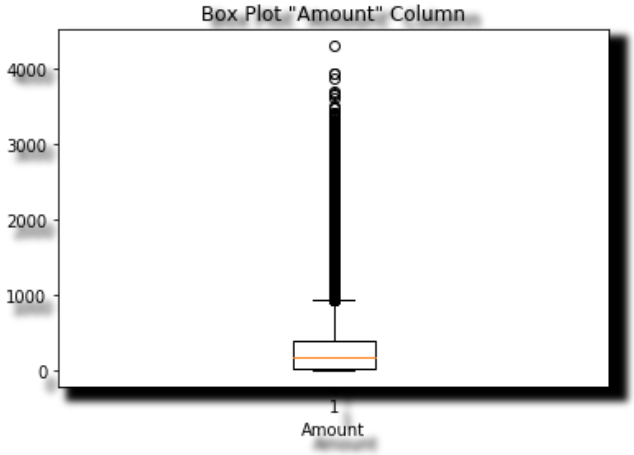
- Design of reshaping:
  - ‘name’: column dropped
  - ‘lastname’: column dropped
  - ‘Notes’: column will be dropped
  - 'Photo_Type' column lowercased & stripped dots
  - ‘Inv_Date' column converted to datetime type for better analysis

- Issues fixed:
  - 2 redundant columns and 1 unrelated column dropped
  - 1 column type converted to datetime
  - 1 column cleaned
  - column names changed / organized

2.3. Statistics summary – describe( )

Numeric feature of interest (Amount)

```
count    98572.000000
mean       331.195788
std        427.601356
min         10.030000
25%         36.447500
50%        183.105000
75%        396.105000
max       4301.420000
Name: Amount, dtype: float64
```



Other features

Number of unique Customer IDs: 14414
Number of unique Invoice Numbers: 98572
Number of unique Invoice Dates: 360

Number of unique Photo Types: 7

- Amount:
  - Mean of the sales amount is 331.2. Standard deviation is 427.6. The minimum value is 10.03, and maximum is 4301.42.

- Other features statistics:
  - Number of unique customers being less than number of rows shows that there are returning customers.
  - Invoice numbers are equal to number of rows (no duplications).
  - Invoice dates cover around a year.
  - There are 7 unique photo types.

3. COHORT ANALYSIS

Cohort Analysis is a descriptive analytics tool that groups customers into segments, i.e. cohorts, which are then measured over time.

Analyzing time cohorts shows the customer behavior depending on the time they started using the company's products / services.

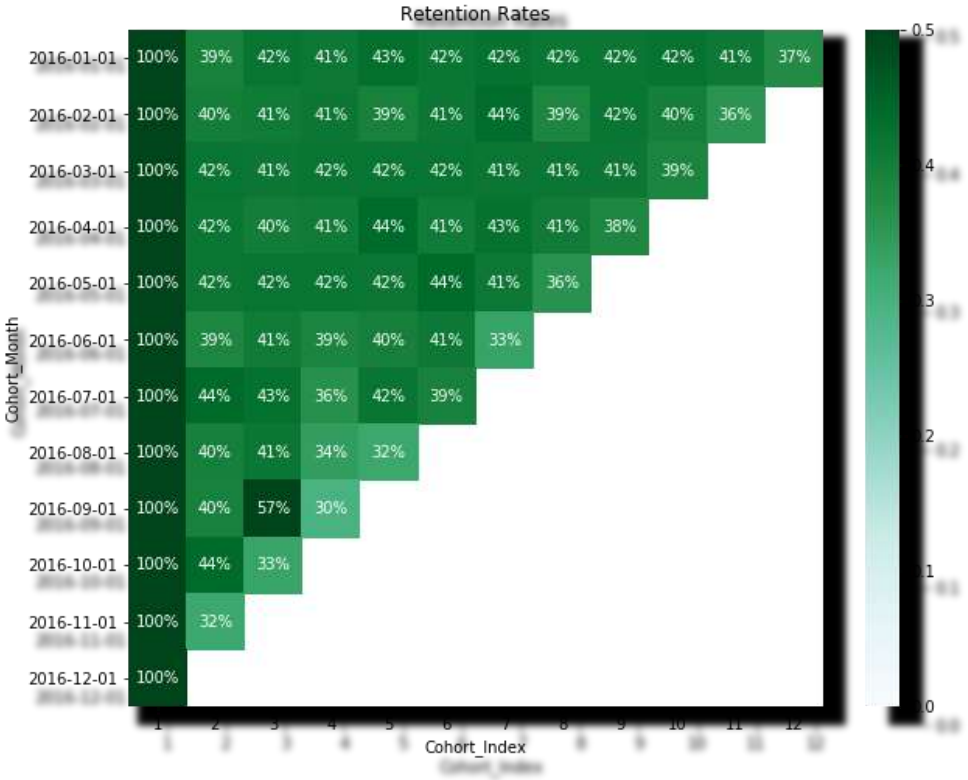
To facilitate, I extracted month values from the data to do calculations. I added 'Inv_Month' as a new column showing which month the transaction was made. Another column added is the 'Cohort_Month' column, which shows the month of the first purchase of the customer. Here are some random rows of the new DataFrame:

	Inv_No	Inv_Date	Cust_ID	Photo_Type	Amount	Inv_Month	Cohort_Month
97908	141799	2016-12-24	14809	official	1343.37	2016-12-01	2016-01-01
52352	96243	2016-07-11	14567	self_taken	43.66	2016-07-01	2016-02-01
79095	122986	2016-10-16	5463	official	196.69	2016-10-01	2016-02-01
17981	61872	2016-03-07	6504	wedding	322.85	2016-03-01	2016-03-01
90233	134124	2016-11-26	4367	wedding	326.58	2016-11-01	2016-01-01

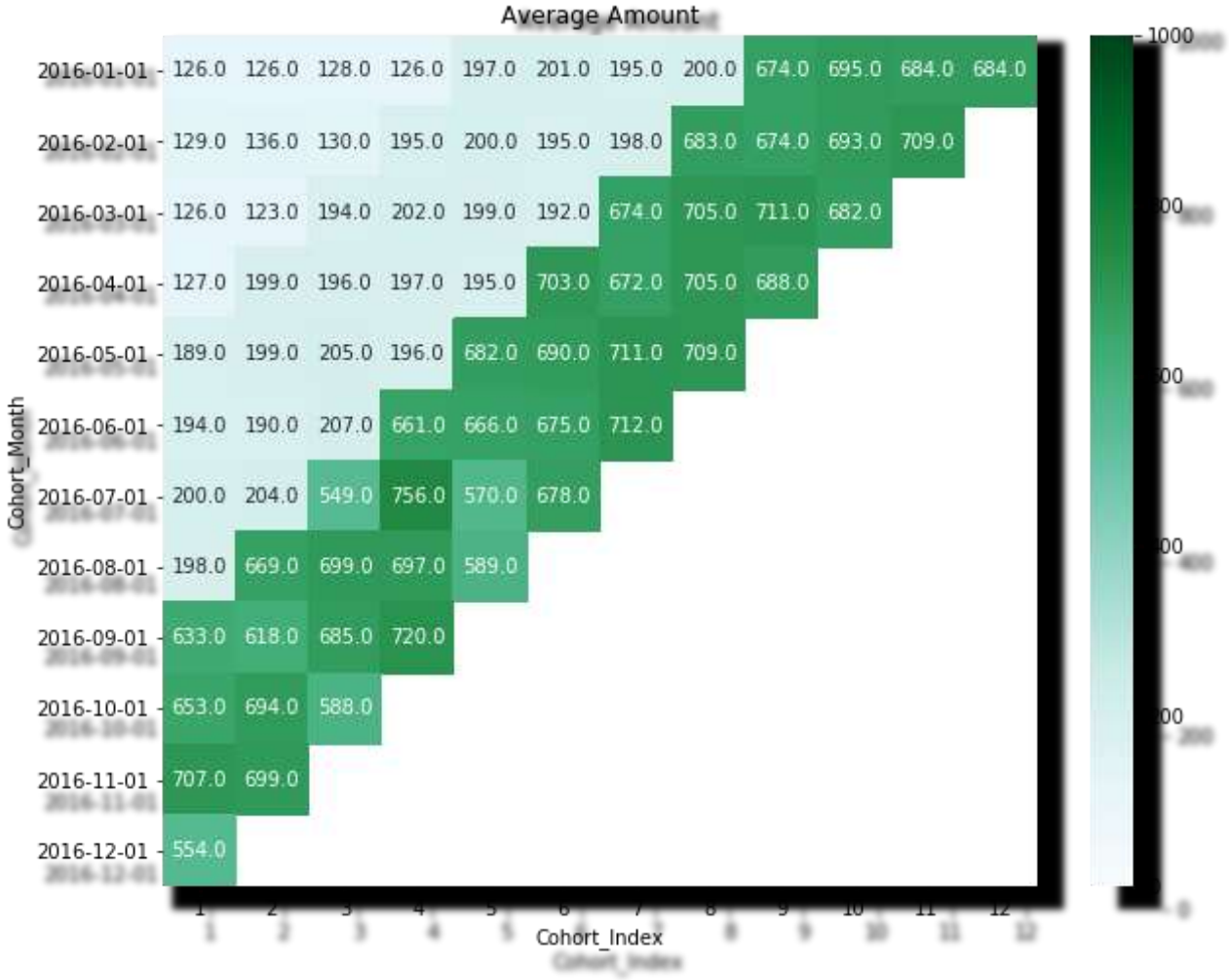
Next, I calculated the number of monthly active customers from each cohort. To do this, I defined a function to extract year, month and day integer values. Then, I used the function to calculate time offset value to find how many months passed since the first purchase.

Cohort_Index	1	2	3	4	5	6	7	8	9	10	11	12
Cohort_Month												
2016-01-01	6115.0	2397.0	2578.0	2514.0	2633.0	2551.0	2599.0	2544.0	2542.0	2587.0	2519.0	2285.0
2016-02-01	3242.0	1300.0	1323.0	1338.0	1274.0	1337.0	1411.0	1251.0	1358.0	1293.0	1159.0	NaN
2016-03-01	2093.0	880.0	863.0	878.0	880.0	881.0	867.0	867.0	865.0	810.0	NaN	NaN
2016-04-01	1179.0	491.0	476.0	479.0	517.0	479.0	502.0	480.0	453.0	NaN	NaN	NaN
2016-05-01	708.0	294.0	298.0	297.0	294.0	311.0	291.0	257.0	NaN	NaN	NaN	NaN
2016-06-01	415.0	160.0	169.0	161.0	165.0	171.0	137.0	NaN	NaN	NaN	NaN	NaN
2016-07-01	256.0	112.0	109.0	93.0	107.0	101.0	NaN	NaN	NaN	NaN	NaN	NaN
2016-08-01	179.0	71.0	74.0	61.0	58.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-09-01	111.0	44.0	63.0	33.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-10-01	64.0	28.0	21.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-11-01	28.0	9.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2016-12-01	14.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Above, the row values represent the cohort (the month of the first purchase). The column values represent months since acquisition. Entries in the table are count of active customers. The first column shows the total number of customers making their first purchase in that month (cohort). So, I divided all entries in a row by the first column to get percentages of how many of them are retained. The result is the retention rate table. I used seaborn to plot a heatmap of the retention table.



Similarly, I built another cohort table where entries are monthly average amounts and visualized as a heatmap:



While customer retention rate remains relatively constant around 40%, there is an increase in sales towards Christmas.

To identify, verify and visualize segments in the data, I used several different methods. Namely,

- RFMT analysis,
- K-Means clustering,
    - Snake plots,
    - Relative Importance of Segment Attributes heatmap,
    - Silhouette scores.

4. RFMT Analysis

RFMT stands for Recency, Frequency, Monetary Value and Tenure.

- Recency measures how recent the last transaction of each customer is.
- Frequency measures how many transactions each customer did in the year.
- Monetary Value measures the amount each customer spent in the year.
- Tenure measures how much time passed since the first transaction of each customer.

To calculate recency, I used a snapshot date as one day later after the last purchase in the DataFrame. For each customer,

- I calculated days passed between snapshot_date and last purchase (Recency),
- counted invoices for frequency metric (Frequency),
- summed all the spent amount (Monetary Value),
- calculated days passed between snapshot_date and first purchase (Tenure) and saved as a new DataFrame:

Cust_ID	Recency	Frequency	Monetary_Value	Tenure
2561	54	4	3516.27	361
2562	15	6	1239.98	270
2563	22	6	1916.68	361
2564	6	9	4339.82	301
2565	163	5	967.33	330

Here, a low recency value is better. It shows that the customer has been to the company recently.

4.1. Building RFMT Segments and RFMT Score

Next, I segmented the customers into 4 equal size quartiles using pandas 'qcut()' function and calculated an RFMT score for each customer based on the quartile.

Cust_ID	Recency	Frequency	Monetary_Value	Tenure	R_qrtl	F_qrtl	M_qrtl	T_qrtl	RFMT_Segment	RFMT_Score
2561	54	4	3516.27	361	2	1	4	4	2144	11.0
2562	15	6	1239.98	270	4	2	2	1	4221	9.0
2563	22	6	1916.68	361	3	2	3	4	3234	12.0
2564	6	9	4339.82	301	4	4	4	2	4442	14.0
2565	163	5	967.33	330	1	1	1	3	1113	6.0

4.2. Grouping by the RFMT Score:

RFMT_Score	Recency mean	Frequency mean	Monetary_Value mean	Monetary_Value count	Tenure mean
4.0	143.7	2.6	487.7	479	226.3
5.0	112.8	3.4	762.7	558	251.6
6.0	106.6	3.8	922.9	978	278.1
7.0	102.9	4.1	1013.1	1370	304.8
8.0	65.7	4.9	1473.9	1328	302.5
9.0	56.3	5.4	1720.4	1518	313.3
10.0	48.6	6.1	1975.6	1495	322.1
11.0	41.6	6.8	2266.1	1481	328.4
12.0	36.2	7.5	2571.1	1479	335.1
13.0	28.4	8.1	2940.3	1327	341.5
14.0	22.7	9.0	3375.4	1151	347.5
15.0	15.6	9.7	3742.5	804	355.2
16.0	8.4	10.7	4389.9	436	361.0

By looking at the table above, I did 2 different classifications:

- I classified customers with an RFMT score >= 12 as Gold members, and RFMT Score <12 as Silver members,
- I classified customers with an RFMT score >= 12 as Gold members, RFMT Score >= 9 as Silver members and RFMT Score < 9 as Bronze Members:

_Segment	Recency mean	Frequency mean	Monetary_Value mean	Monetary_Value count	Tenure mean
1.Gold	25.7	8.6	3177.3	5197	344.8
2.Silver	74.3	5.0	1501.9	9207	302.3

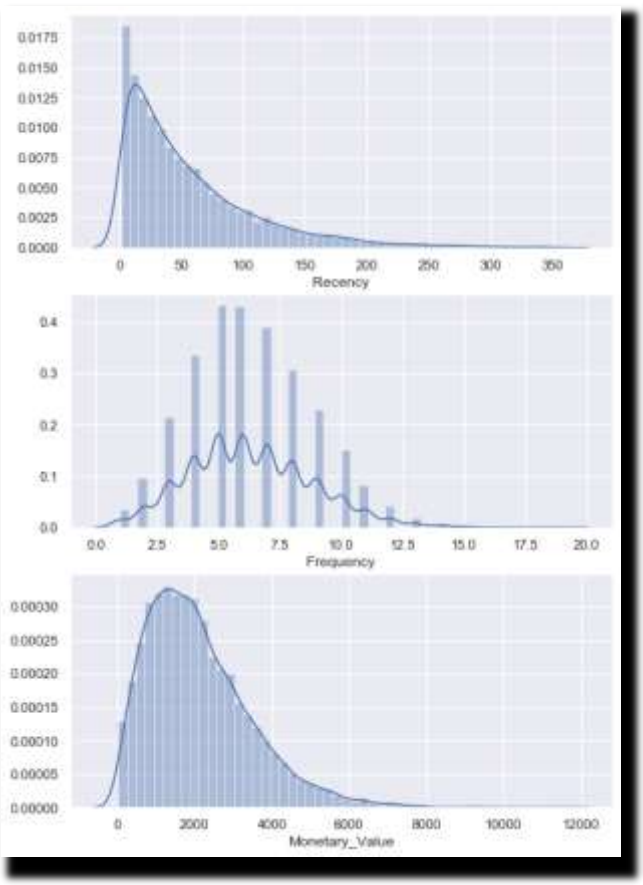
Member_Segment	Recency mean	Frequency mean	Monetary_Value mean	Monetary_Value count	Tenure mean
1.Gold	25.7	8.6	3177.3	5197	344.8
2.Silver	48.9	6.1	1985.1	4494	321.2
3.Bronze	98.5	4.0	1041.2	4713	284.3

The first segment remained the same in both solutions. The second segment of the 2-segment solution on the left gave birth to 2 other segments in the 3-segment solution on the right.

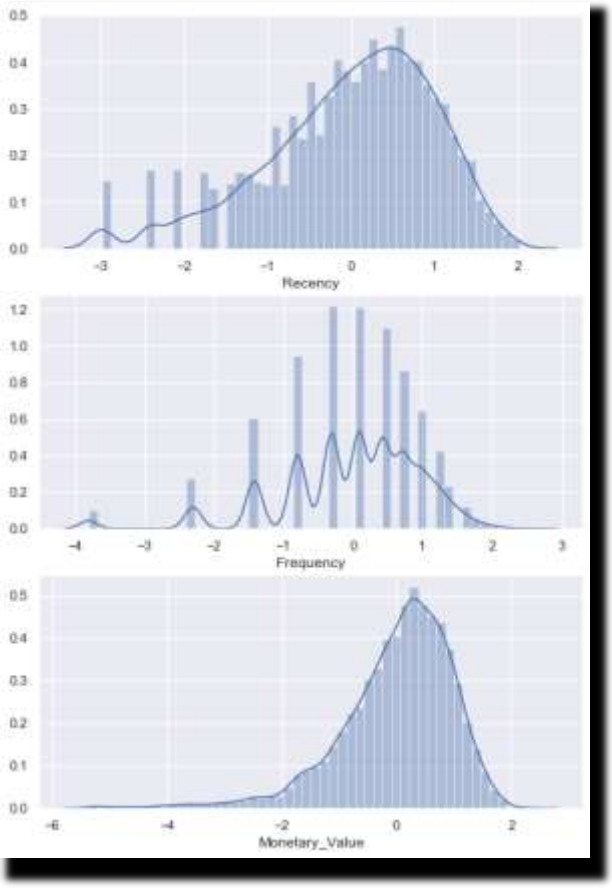
5. K MEANS CLUSTERING

K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. To be able to do K-means clustering, the variables must be symmetrically distributed (i.e. not skewed), and they must have the same mean and variance.

The Recency, Frequency and Monetary Value plots showed that the variables are not symmetrically distrubuted. All of the plots were right skewed. To mitigate, I used the log transformation to unskew the data:



Distributions not symmetrical



Applied Log transformation

Checking the mean and standard deviation with describe() method, Summary statistics showed that the variables had different means and variances. Thus, I standardized the variables to the same mean and scaled to the same standard deviation.

	Recency	Frequency	Monetary_Value
count	14404.000000	14404.000000	14404.000000
mean	56.743266	6.329700	2106.415806
std	54.842904	2.523585	1350.573898
min	1.000000	1.000000	20.030000
25%	17.000000	5.000000	1099.515000
50%	40.000000	6.000000	1871.580000
75%	79.000000	8.000000	2860.025000
max	357.000000	19.000000	11579.080000

	Recency	Frequency	Monetary_Value
count	1.440400e+04	1.440400e+04	1.440400e+04
mean	-5.440124e-17	7.571777e-16	-5.279571e-16
std	1.000035e+00	1.000035e+00	1.000035e+00
min	-2.984293e+00	-3.807992e+00	-5.392941e+00
25%	-5.738345e-01	-3.112319e-01	-4.863011e-01
50%	1.541542e-01	8.489076e-02	1.652976e-01
75%	7.331723e-01	7.099259e-01	6.847591e-01
max	2.016403e+00	2.589271e+00	2.397773e+00

After preprocessing the data, I started clustering by using the Elbow criterion method to select a good value for k. I plotted the number of clusters (k) against within-cluster-sum-of-squared-errors (SSE), i.e. sum of squared distances from every data point to its cluster center. The elbow point in the plot (where the decrease in SSE slows down) and the next point are good candidates for the best k.

The elbow lies at k = 2. I also tried k = 3 and k = 4.

Cluster	Recency mean	Frequency mean	Monetary_Value mean	count
0	101.0	4.0	952.0	4994
1	34.0	8.0	2719.0	9410

- With k=2, the clusters were far more distinct. But cluster 1 was almost double the size of cluster 0.

Cluster	Recency mean	Frequency mean	Monetary_Value mean	count
0	112.0	3.0	699.0	3126
1	10.0	8.0	2972.0	4035
2	59.0	7.0	2232.0	7243

- With k=3, the Frequency and Monetary Value of Clusters 0 and 2 seemed somewhat close.

Cluster	Recency mean	Frequency mean	Monetary_Value mean	count
0	7.0	7.0	2699.0	2966
1	143.0	3.0	394.0	1331
2	76.0	5.0	1335.0	4984
3	45.0	8.0	2959.0	5123

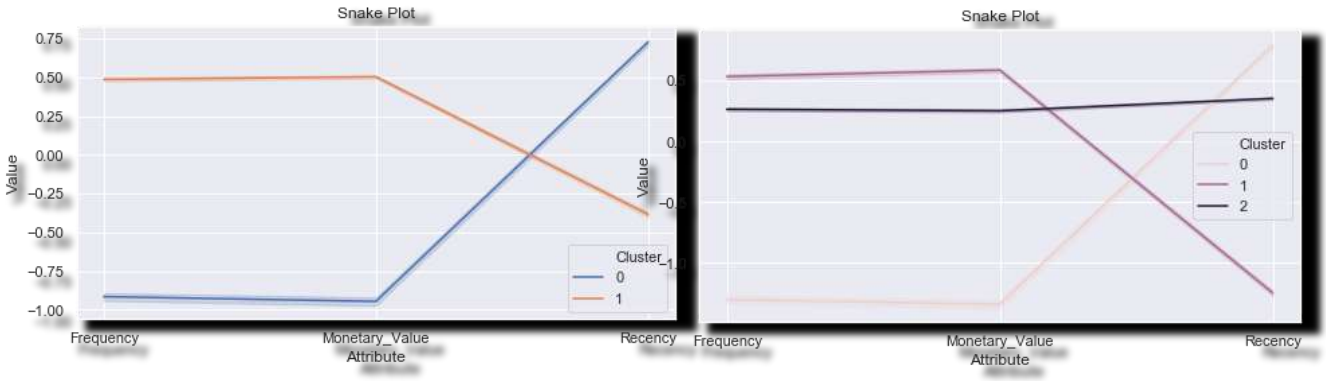
- With k=4, The Frequency and Monetary Value of Clusters 0 and 3 seemed very close.

So, K-Means clustering favors k=2. However, k=3 clusters is still an option to balance the simplicity and gaining more insights.

This is also in line with the RFM analysis. To verify and visualize the number of K-Means clusters, I used snake plots, relative importance of segment attributes and silhouette analysis.

5.1. Snake Plots

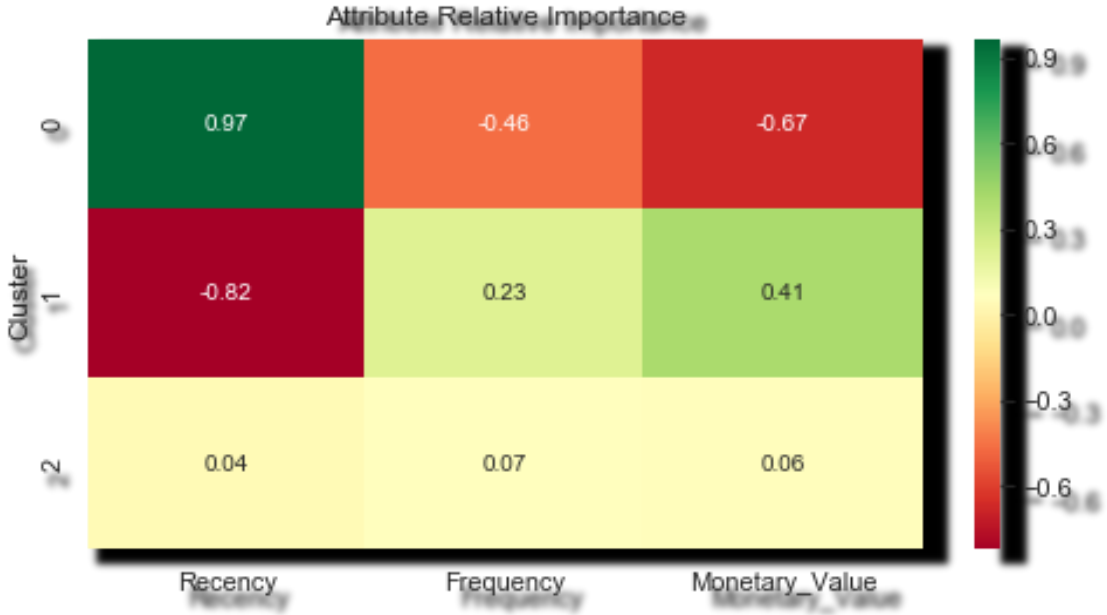
I used snake plots to visualize the clustering. To do this, I first transformed the 'RFM_normalized' back to a DataFrame structure which became a numpy ndarray object after scaling. Second, I added a cluster column and melted the data for easier plotting.



Snake plots are a visual way to identify / verify clusters. Here, there is some overlap with 3 clusters.

5. 2. Relative Importance of Segment Attributes

Next, I looked at the relative importance of segment attributes as a proportion of cluster average to population average and visualized as a heatmap. Here, the further a ratio is from 0, the more important that attribute is for defining a specific cluster compared to the population average.



The Relative Importance of Segment Attributes heatmap indicates none of the attributes are important for defining Cluster 2 compared to the population average.

5.3. Silhouette Analysis

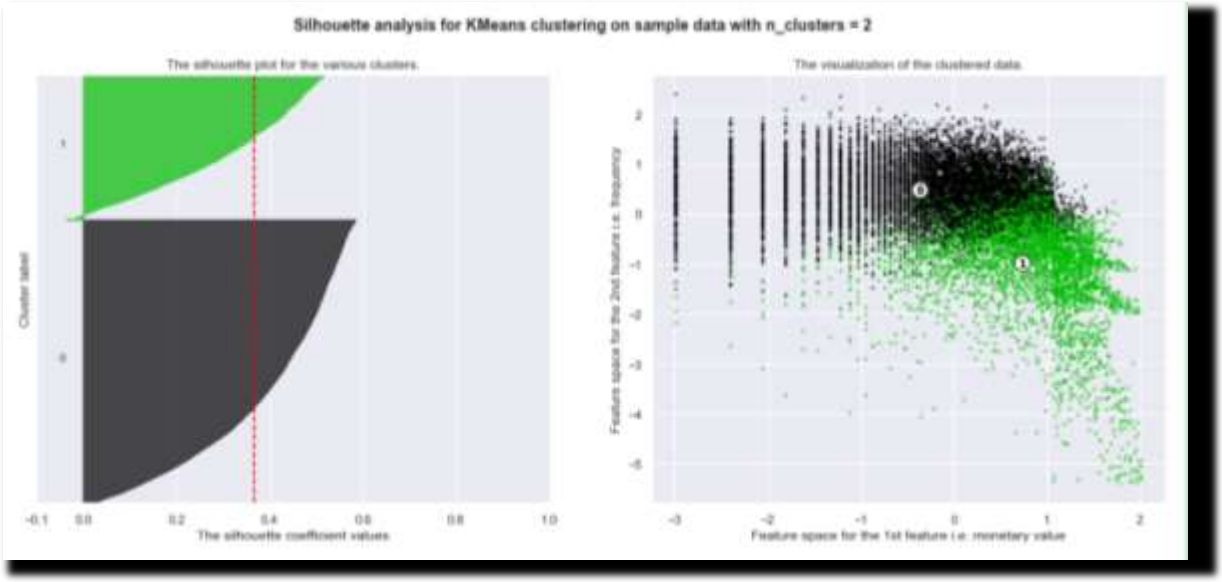
Finally, I looked at the silhouette scores to verify the clustering. This is a method that measures how well each datapoint x_i "fits" its assigned cluster and also how poorly it fits into other clusters. This is a different way of looking at the same objective. Denote ax_i as the average distance from x_i to all other points within its own cluster k. The lower the value, the better. On the other hand, bx_i is the minimum average distance from x_i to points in a different cluster, minimized over clusters. That is, compute separately for each cluster the average distance from x_i to the points within that cluster, and then take the minimum. The silhouette $s(x_i)$ is defined as seen on the right.

$$s(x_i) = \frac{b_{x_i} - a_{x_i}}{\max\left(a_{x_i}, b_{x_i}\right)}$$

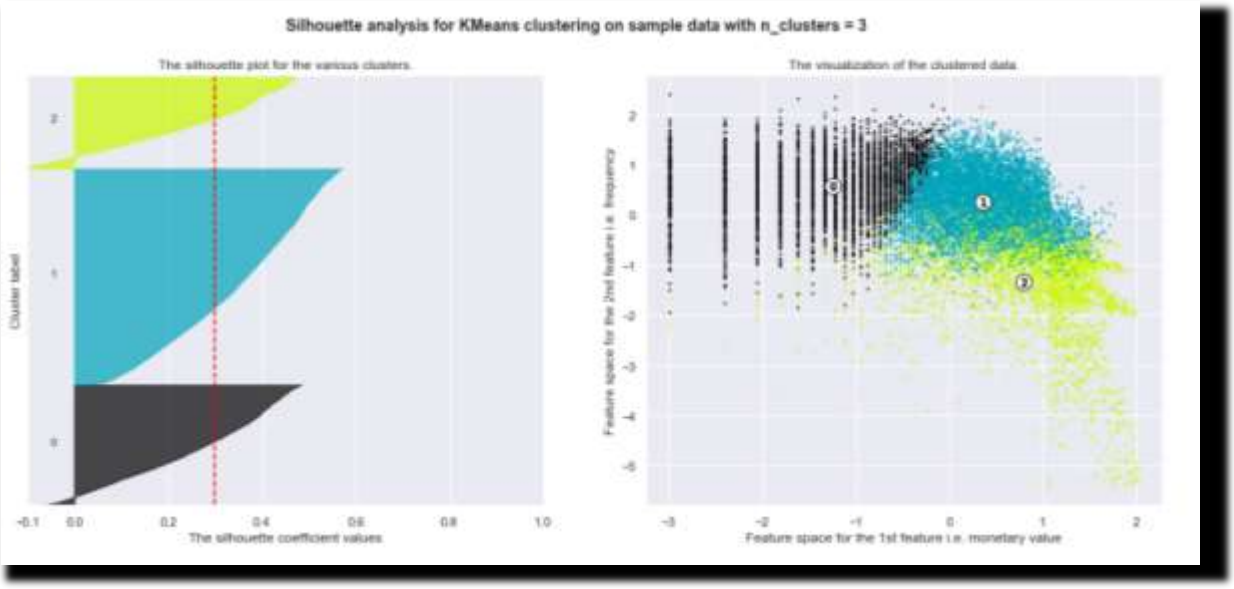
Range	Interpretation
0.71 - 1.0	A strong structure has been found.
0.51 - 0.7	A reasonable structure has been found.
0.26 - 0.5	The structure is weak and could be artificial.
< 0.25	No substantial structure has been found.

The silhouette score is computed on every datapoint in every cluster. The silhouette score ranges from -1 (a poor clustering) to +1 (a very dense clustering) with 0 denoting the situation where clusters overlap. Some criteria for the silhouette coefficient is provided in the table on the left.

The silhouette score with n_clusters =2 was around 0.37. The visualization is shown below:



The silhouette score with n_clusters =3 was around 0.3. The visualization is shown below:



So, best silhouette score was obtained with n-clusters=2. However, according to the criteria for the silhouette coefficient score of around 0.37, the structure is weak and could be artificial.

6. CONCLUSION

Overall, I used several methods to do, verify and visualize customer segmentation:

- RFMT Analysis,
- K-Means clustering,
  - Snake Plots,
  - Relative Importance of Segment Attributes,
  - Silhouette scores.

- All methods clearly identified two clusters in the data.
- While favoring 2 clusters, a 3 clusters option was also possible in all methods.
- Through RFMT Analysis, I was able to perform a fair segmentation into 3 almost equally distanced clusters in terms of RFMT and size of each clusters.
- For better capturing customer behavior and more focused marketing targeting diverse customers, I would suggest using a 3-clustered customer segmentation pending managerial decision.