PRML Minor Project

Project 4

Q: A company that sells some of the product, and you want to know how well the selling performance of the product. You have the data that we can analyze, but what kind of analysis can we do? Well, we can segment customers based on their buying behavior on the market. Your task is to classify the data into the possible types of customers which the retailer can encounter.

The data used is: 'https://archive.ics.uci.edu/ml/datasets/online+retail'

Soln:

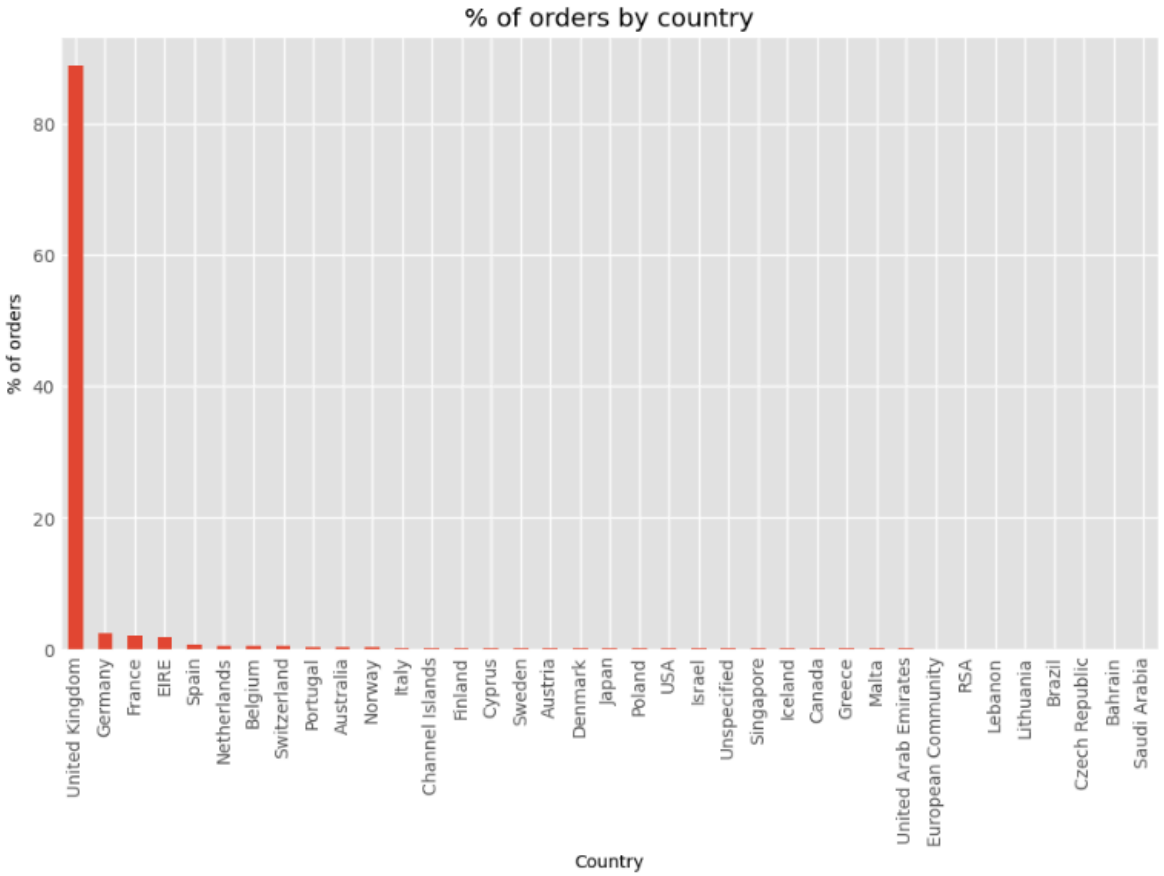
Data analysis and preprocessing:

For this project we started by analyzing the data. The data was huge with more than 5 lakh entries divided over different countries.

Firstly we Preprocessed the data, imported the data as csv, then removed the missing, duplicate and na values. Then we dropped the Description column as it was not in use, other columns like quantity and unit price already described the type of order so there was no need of order description.

Then for analyzing more we find out the no. of unique customers which came out as 4372, no. of unique transactions which came out to be 22190 and unique products 3684.

Then we thought and divided the data into different countries and made a bar graph for it. From that graph we found out that United Kingdom has maximum no. of orders with more than 95% of orders. So then we thought of analyzing it for UK first.



After setting the data for only UK we first found out the total no. of cancelled orders which came out as 7501 in number.

After removing all the false, duplicate, NA, and cancelled orders the data left was about 3.5 lakh which was earlier close to 6 lakh.

Now the Unique customer = 3921,

Unique transactions = 16649,

Unique products = 3645

After preprocessing we first created columns for invoice month. The main goal for this was to find out how old each customer is so the first month of their order is noted.

For analyzing we thought to use Cohort analysis.

Cohort analysis involves grouping of data based on their common characteristics.

Cohort analysis based on the month of order involves grouping customers who made their first purchase in the same month and analyzing their behavior over time. This type of analysis can help to understand how customers who joined the business at the same time are behaving, and how they differ from other cohorts.

For cohort analysis, there were few labels that we created:

1) Invoice period: A string representation of the year and month of a single transaction/invoice.
2) Cohort group: A string representation of the year and month of the customer's first purchase. This label is common across all invoices for a particular customer.
3) Cohort period / Cohort Index: A integer representation a customer's stage in its "lifetime". The number represents the number of months passed since the first purchase.

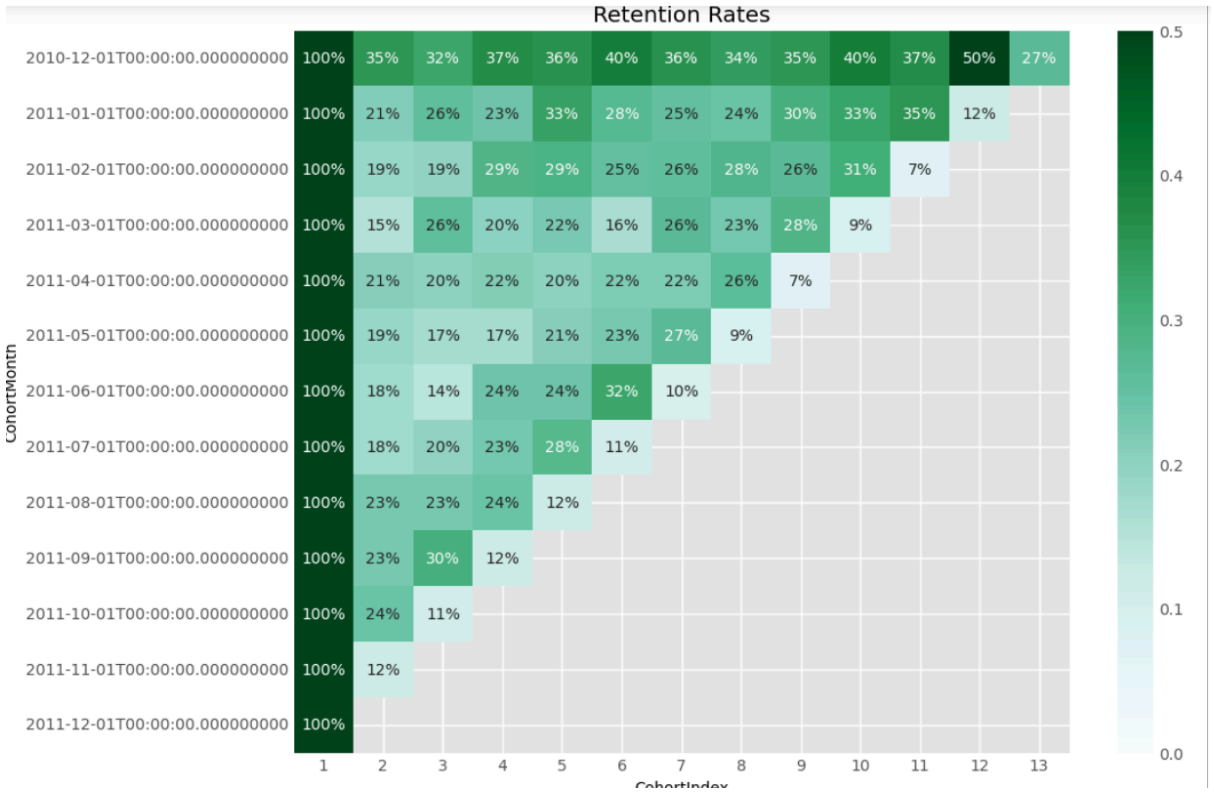
	CustomerID	InvoiceMonth
0	12346	2011-01-01
1	12747	2010-12-01
2	12748	2010-12-01
3	12749	2011-05-01
4	12820	2011-01-01
...	...	...
3916	18280	2011-03-01
3917	18281	2011-06-01
3918	18282	2011-08-01
3919	18283	2011-01-01
3920	18287	2011-05-01

CohortIndex	1	2	3	4	5	6	7	8	9	\
CohortMonth										
2010-12-01	815.0	289.0	263.0	304.0	293.0	323.0	291.0	278.0	289.0	
2011-01-01	358.0	76.0	93.0	84.0	119.0	99.0	90.0	87.0	108.0	
2011-02-01	340.0	64.0	66.0	97.0	98.0	86.0	87.0	96.0	90.0	
2011-03-01	419.0	64.0	109.0	83.0	94.0	69.0	111.0	96.0	119.0	
2011-04-01	277.0	58.0	56.0	60.0	56.0	61.0	61.0	73.0	20.0	
2011-05-01	256.0	48.0	44.0	44.0	53.0	58.0	68.0	23.0	NaN	
2011-06-01	214.0	38.0	31.0	51.0	51.0	69.0	21.0	NaN	NaN	
2011-07-01	169.0	30.0	33.0	39.0	47.0	18.0	NaN	NaN	NaN	
2011-08-01	141.0	32.0	32.0	34.0	17.0	NaN	NaN	NaN	NaN	
2011-09-01	276.0	63.0	83.0	32.0	NaN	NaN	NaN	NaN	NaN	
2011-10-01	324.0	79.0	36.0	NaN	NaN	NaN	NaN	NaN	NaN	
2011-11-01	298.0	35.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2011-12-01	34.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

CohortIndex	10	11	12	13
CohortMonth				
2010-12-01	325.0	299.0	405.0	218.0
2011-01-01	117.0	127.0	43.0	NaN
2011-02-01	104.0	25.0	NaN	NaN
2011-03-01	38.0	NaN	NaN	NaN
2011-04-01	NaN	NaN	NaN	NaN
2011-05-01	NaN	NaN	NaN	NaN
2011-06-01	NaN	NaN	NaN	NaN
2011-07-01	NaN	NaN	NaN	NaN
2011-08-01	NaN	NaN	NaN	NaN
2011-09-01	NaN	NaN	NaN	NaN
2011-10-01	NaN	NaN	NaN	NaN
2011-11-01	NaN	NaN	NaN	NaN
2011-12-01	NaN	NaN	NaN	NaN

	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	InvoiceMonth	CohortMonth	CohortIndex
0	536365	85123A	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	2010-12-01	2010-12-01	1
1	536365	71053	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	2010-12-01	2010-12-01	1
2	536365	84406B	8	2010-12-01 08:26:00	2.75	17850	United Kingdom	2010-12-01	2010-12-01	1
3	536365	84029G	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	2010-12-01	2010-12-01	1
4	536365	84029E	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	2010-12-01	2010-12-01	1
...	...	...	...	...	...	...	...	...	...	...
349222	581585	22466	12	2011-12-09 12:31:00	1.95	15804	United Kingdom	2011-12-01	2011-05-01	8
349223	581586	22061	8	2011-12-09 12:49:00	2.95	13113	United Kingdom	2011-12-01	2010-12-01	13
349224	581586	23275	24	2011-12-09 12:49:00	1.25	13113	United Kingdom	2011-12-01	2010-12-01	13
349225	581586	21217	24	2011-12-09 12:49:00	8.95	13113	United Kingdom	2011-12-01	2010-12-01	13
349226	581586	20685	10	2011-12-09 12:49:00	7.08	13113	United Kingdom	2011-12-01	2010-12-01	13

Then we created a table for retention rates for the customers.



Then we find out customer I'd with recency:

	CustomerID	Recency
count	3921.000000	3921.000000
mean	15561.471563	92.188472
std	1576.823683	99.528995
min	12346.000000	1.000000
25%	14208.000000	18.000000
50%	15569.000000	51.000000
75%	16913.000000	143.000000
max	18287.000000	374.000000

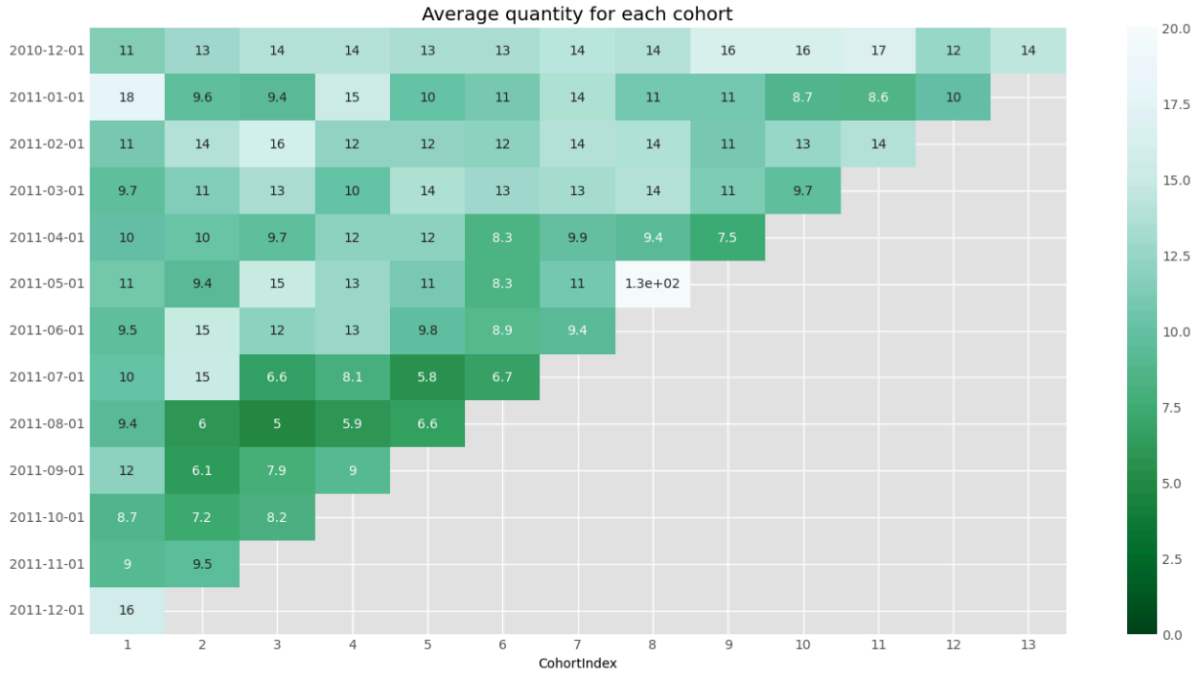
An overview how it will look:

	CustomerID	Recency
0	12346	326.0
1	12747	2.0
2	12748	1.0
3	12749	4.0
4	12820	3.0

Here recency defines no. of days from last purchase.

	Quantity	UnitPrice	CustomerID	CohortIndex	days_since_last_purchase	days_since_last_purchase_num
count	349227.000000	349227.000000	349227.000000	349227.000000	349227	349227.000000
mean	12.181295	2.972124	15548.333625	5.126385	152 days 09:22:10.464139372	151.850642
std	191.797470	17.990307	1594.403077	3.840520	113 days 03:00:44.028768830	113.141808
min	1.000000	0.000000	12346.000000	1.000000	1 days 00:00:00	1.000000
25%	2.000000	1.250000	14191.000000	1.000000	48 days 01:39:00	48.000000
50%	4.000000	1.950000	15518.000000	4.000000	131 days 23:12:00	131.000000
75%	12.000000	3.750000	16931.000000	8.000000	247 days 01:09:00	247.000000
max	80995.000000	8142.750000	18287.000000	13.000000	374 days 04:23:00	374.000000

Then we created heatmap for average quantity for each cohort:



From this analysis we can find how many customers have recent order, which one are the oldest customer, which used to order early and now don't order and which are the new ones to order.

After this we calculated total sum per order by multiplying order quantity and unit price. We renamed this total sum to MonetaryValue.

An overview how the table looks after adding monetary value with recency, customer id and frequency.

CustomerID	Recency	Frequency	MonetaryValue	CustomerID
12346	325	1	77183.60	12346
12747	2	103	4196.01	12747
12748	0	4413	33053.19	12748
12749	3	199	4090.88	12749
12820	3	59	942.34	12820

From we thought of analyzing the data with RFM.

RFM is an acronym of recency, frequency and monetary. Recency is about when was the last order of a customer. It means the number of days since a customer made the last purchase. If it's a case for a website or an app, this could be interpreted as the last visit day or the last login time.

Frequency is about the number of purchases in a given period. It could be 3 months, 6 months or 1 year. So we can understand this value as for how often or how many a customer used the product of a company. The bigger the value is, the more engaged the customers are. Could we say them as out VIP? Not necessary. Cause we also have to think about how much they actually paid for each purchase, which means monetary value.

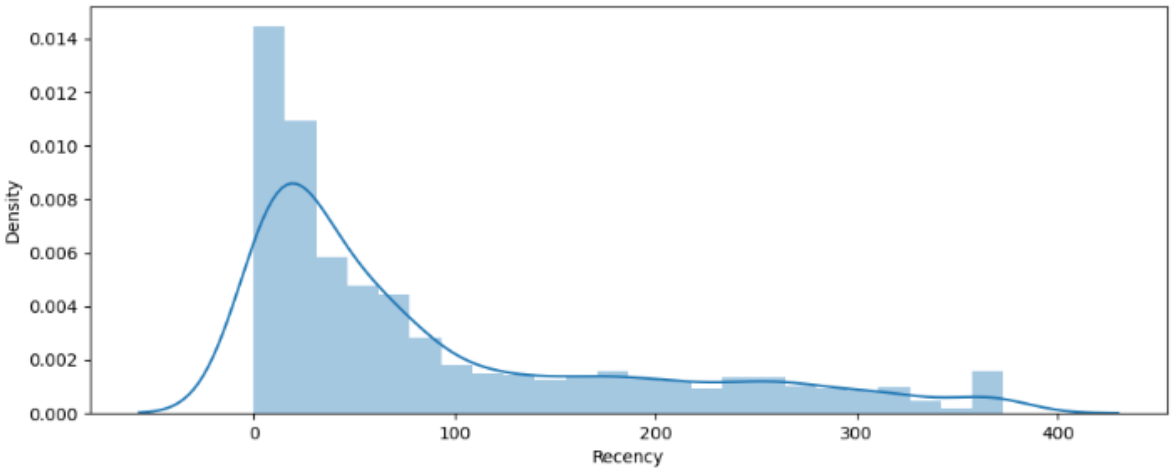
Monetary is the total amount of money a customer spent in that given period. Therefore big spenders will be differentiated with other customers such as MVP or VIP.

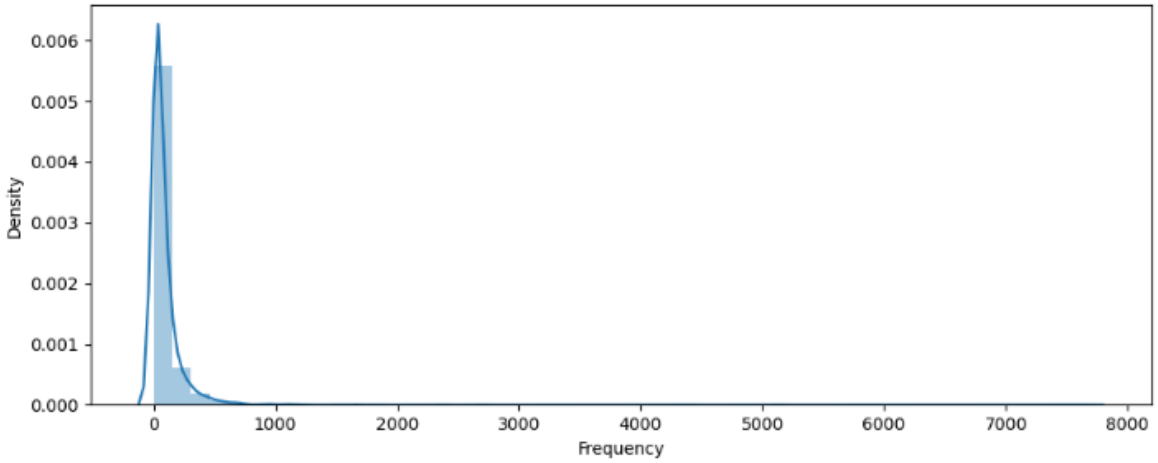
There are 3 RFM metrics, Recency, Frequency, Monetary.

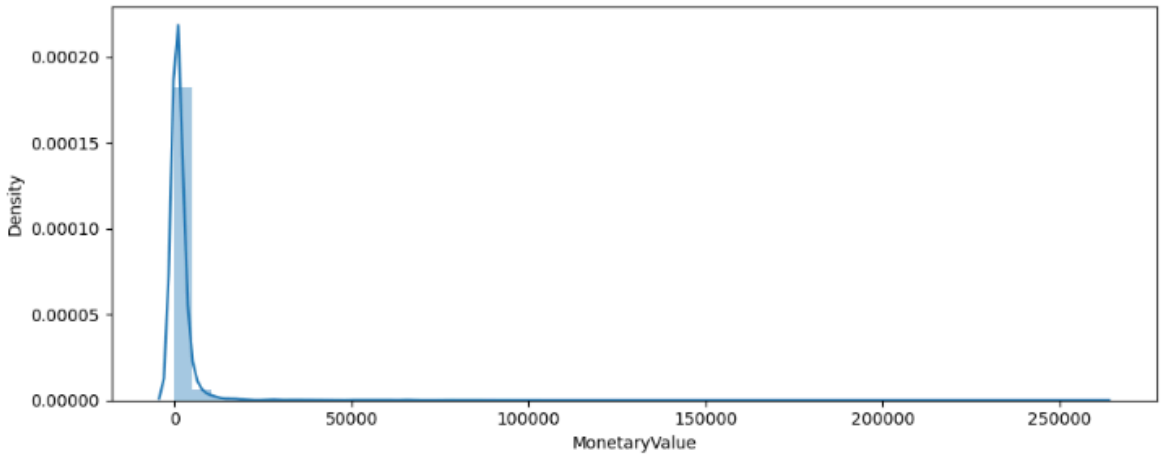
RFM_Score	Recency Mean	Frequency Mean	MonetaryValue Mean	MonetaryValue Count
3	258.11	8.06	151.94	343
4	175.35	13.63	233.51	361
5	151.40	20.54	354.69	471
6	97.04	28.13	823.31	427
7	78.56	38.43	734.08	387

General_Segment	Recency mean	Frequency mean	MonetaryValue mean	MonetaryValue count
Category1	19.4	221.7	4637.3	1142
Category2	71.3	49.1	1051.0	1604
Category3	189.9	14.8	258.3	1175

	Recency	Frequency	MonetaryValue
count	3921.000000	3921.000000	3921.000000
mean	91.722265	89.065800	1857.950687
std	99.528532	214.022733	7477.736186
min	0.000000	1.000000	0.000000
25%	17.000000	17.000000	298.110000
50%	50.000000	40.000000	644.300000
75%	142.000000	98.000000	1570.810000
max	373.000000	7676.000000	259657.300000

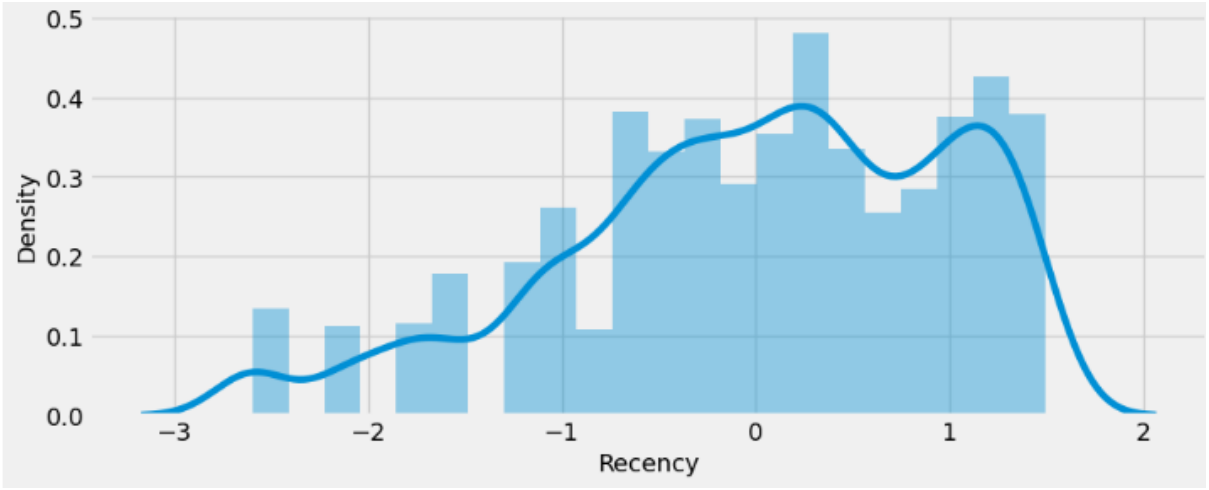

0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
Density
0
100
200
300
400
Recency

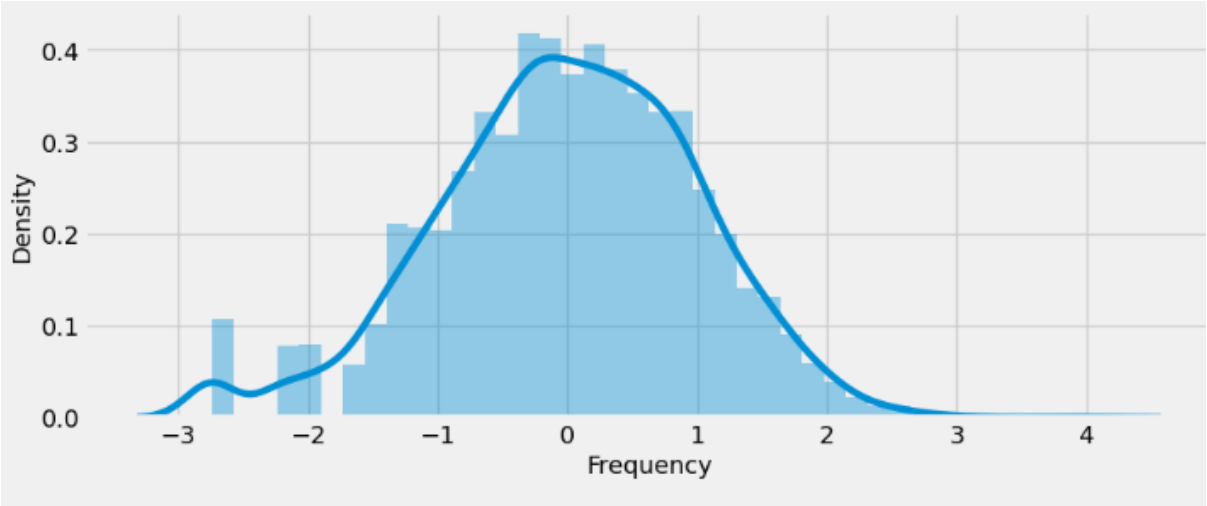

0.006
0.005
0.004
0.003
0.002
0.001
0.000
Density
0
1000
2000
3000
4000
5000
6000
7000
8000
Frequency

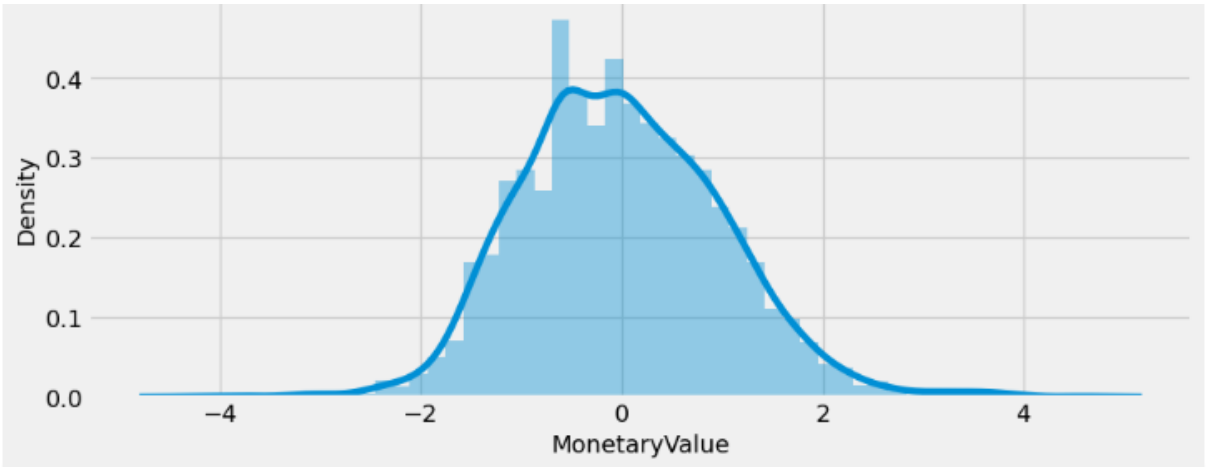

0.00020
0.00015
0.00010
0.00005
0.00000
Density
0
50000
100000
150000
200000
250000
MonetaryValue

	Recency	Frequency	MonetaryValue	CustomerID	RFM_Score
count	3892.000000	3892.000000	3892.000000	3892.000000	3892.000000
mean	92.402107	86.557554	1690.493606	15561.494090	7.464543
std	99.585156	199.436131	5508.790350	1575.889195	2.812486
min	1.000000	1.000000	3.750000	12346.000000	3.000000
25%	17.000000	16.000000	296.170000	14208.750000	5.000000
50%	51.000000	40.000000	639.820000	15567.500000	7.000000
75%	143.250000	97.000000	1544.450000	16910.500000	10.000000
max	373.000000	7676.000000	194390.790000	18287.000000	12.000000

From this table, we find that mean and variance are not equal so for better visualization we standard scaled the data and then created the graphs again.



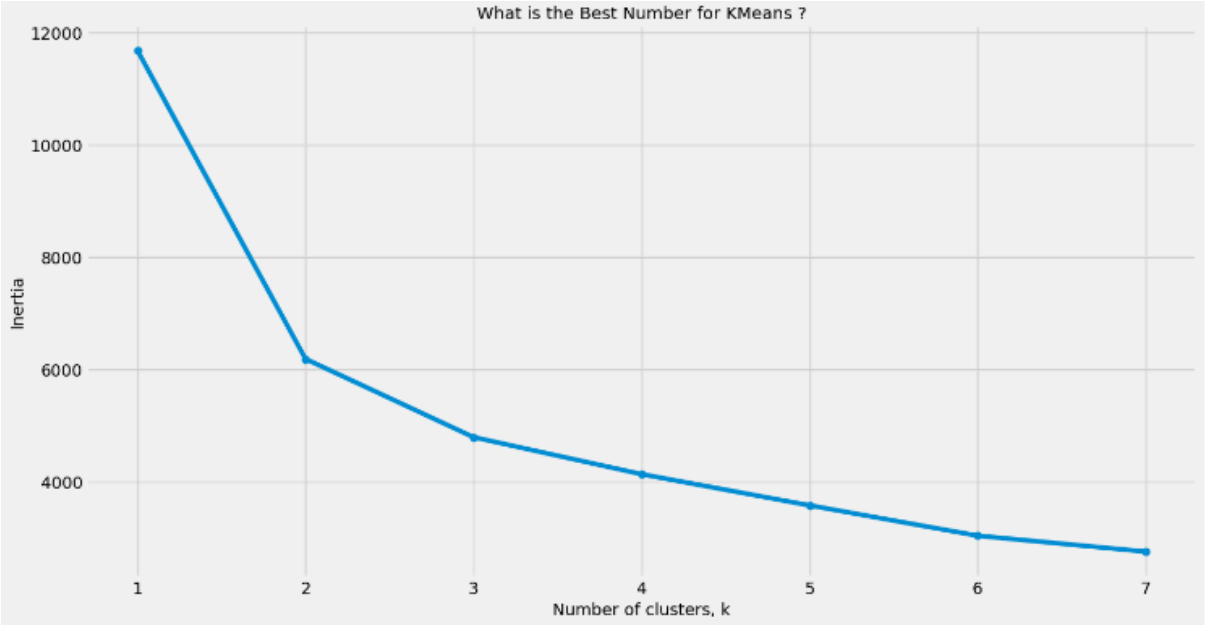




Now we thought of using KMeans clustering for more deeper analysis of the data and solving the problem. For this we created KMeans from scratch.

Key steps:

1) Data pre-processing
2) Choosing a number of clusters
3) Running k-means clustering on pre-processed data
4) Analyzing average RFM values of each cluster



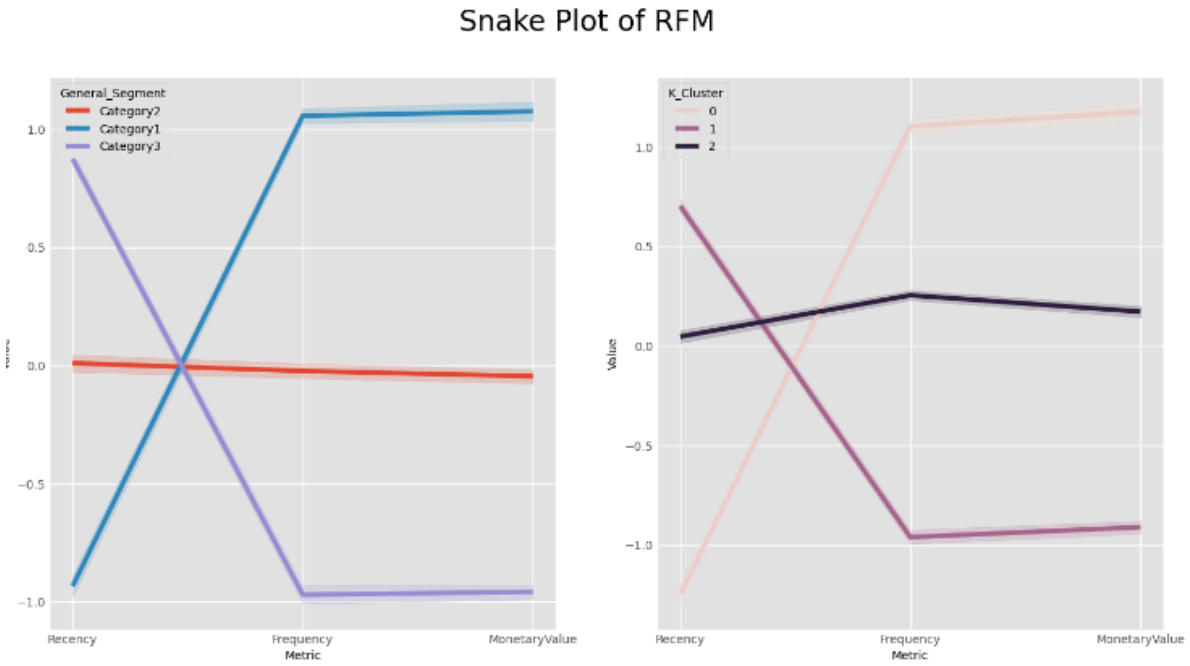
From this graph we used 3 as the number of cluster using the elbow method as it will give the best results.

	Recency	Frequency	MonetaryValue	
	Mean	Mean	Mean	Count
RFM Cluster				
0	11.983626	242.200000	5025.935392	855
1	166.959603	14.815734	283.737669	1411
2	69.989545	66.971710	1157.363163	1626

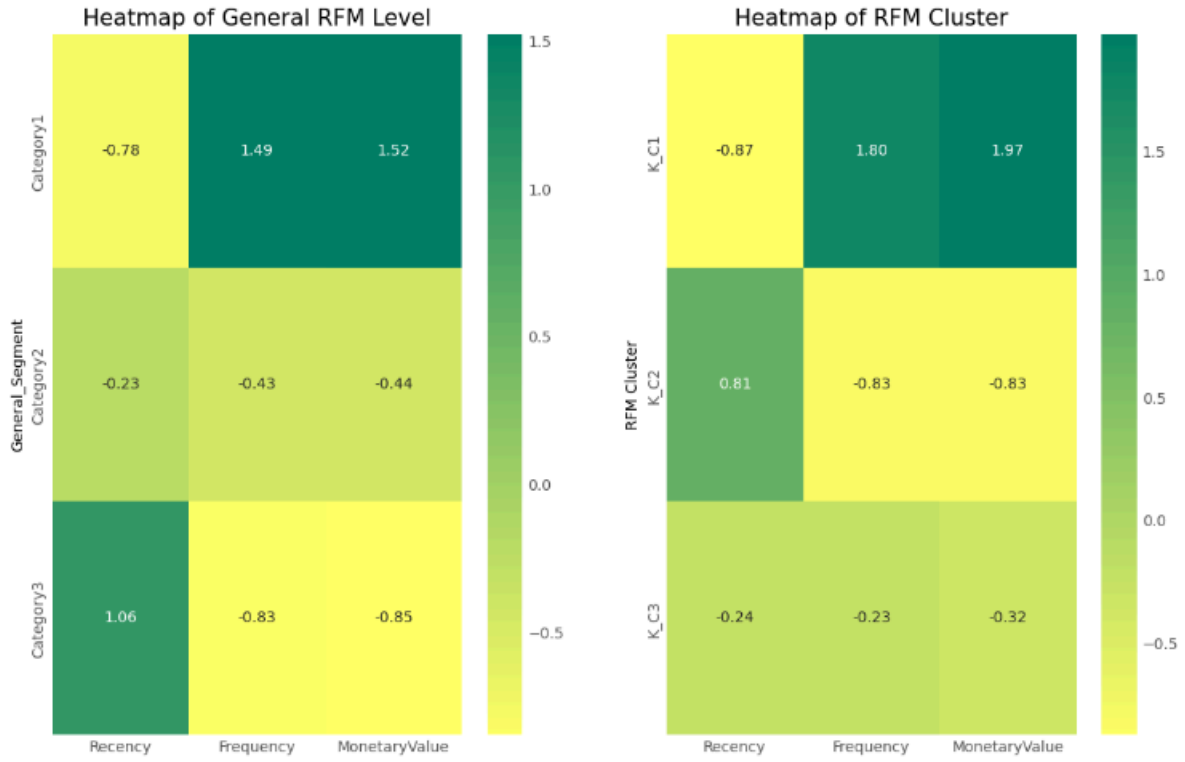
	CustomerID	General_Segment	K_Cluster	Metric	Value
0	12346	Category2	2	Recency	1.399062
1	12747	Category1	0	Recency	-2.121411
2	12749	Category1	0	Recency	-1.841010
3	12820	Category1	0	Recency	-1.841010
4	12821	Category3	1	Recency	1.110098

Then we created snake plots to understand and compare segments

- Market research technique to compare different segments
- Visual representation of each segment's attributes
- Need to first normalize data (center & scale)
- Plot each cluster's average normslized value of each attribute



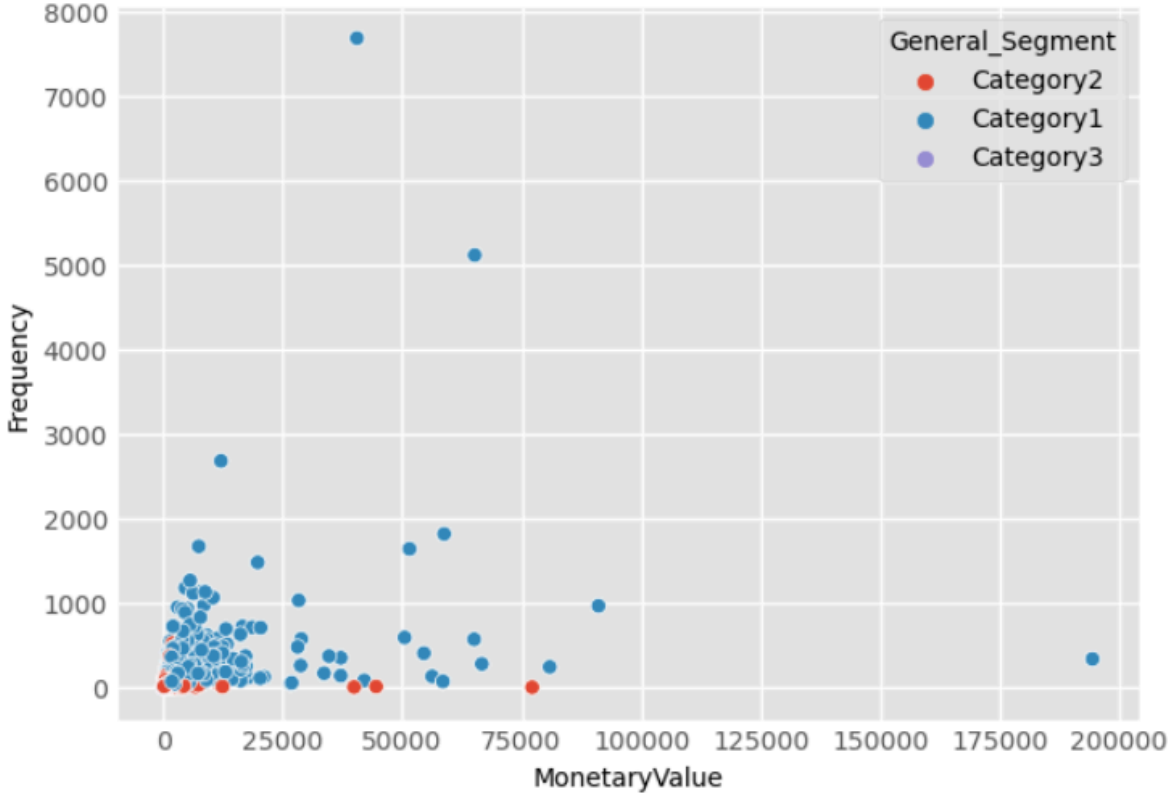
For better understanding we created heat maps. Heat maps are a graphical representation of data where larger values are colored in darker scales and smaller values in lighter. We can compare the variance between groups quite intuitiverly by colors.



From these heat plots we can infer that:

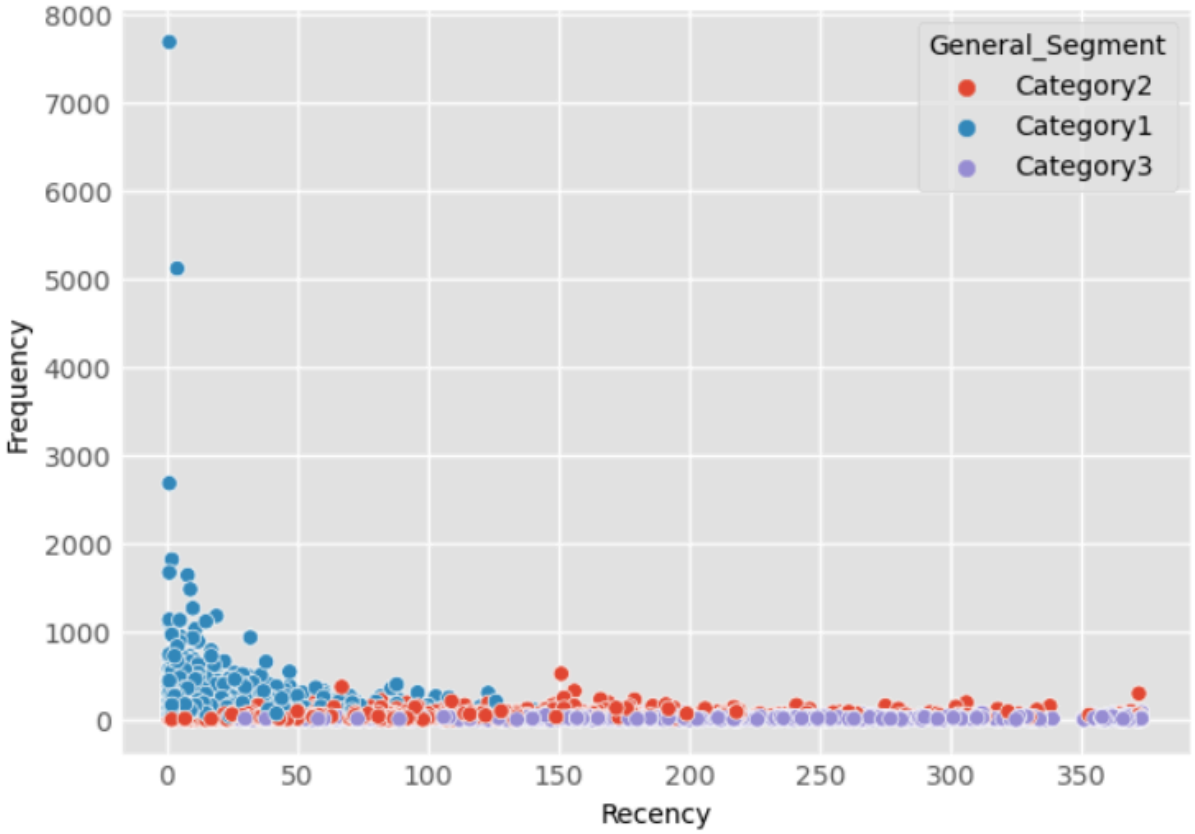
In general RFM level Category 3 has max value for Recency, while in both Requency and Monetary Category 1 has max value.

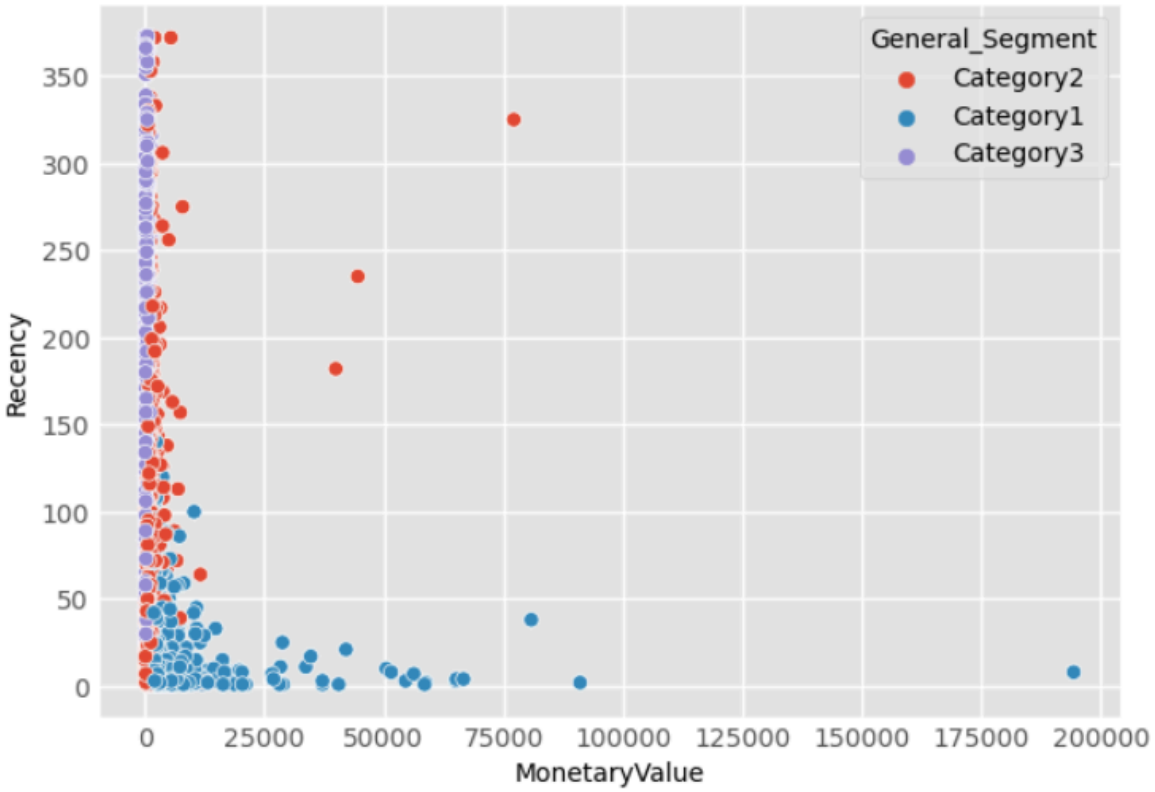
In RFM cluster map, max recency value is in Category 2 while same other 2 values are max in Category 2.



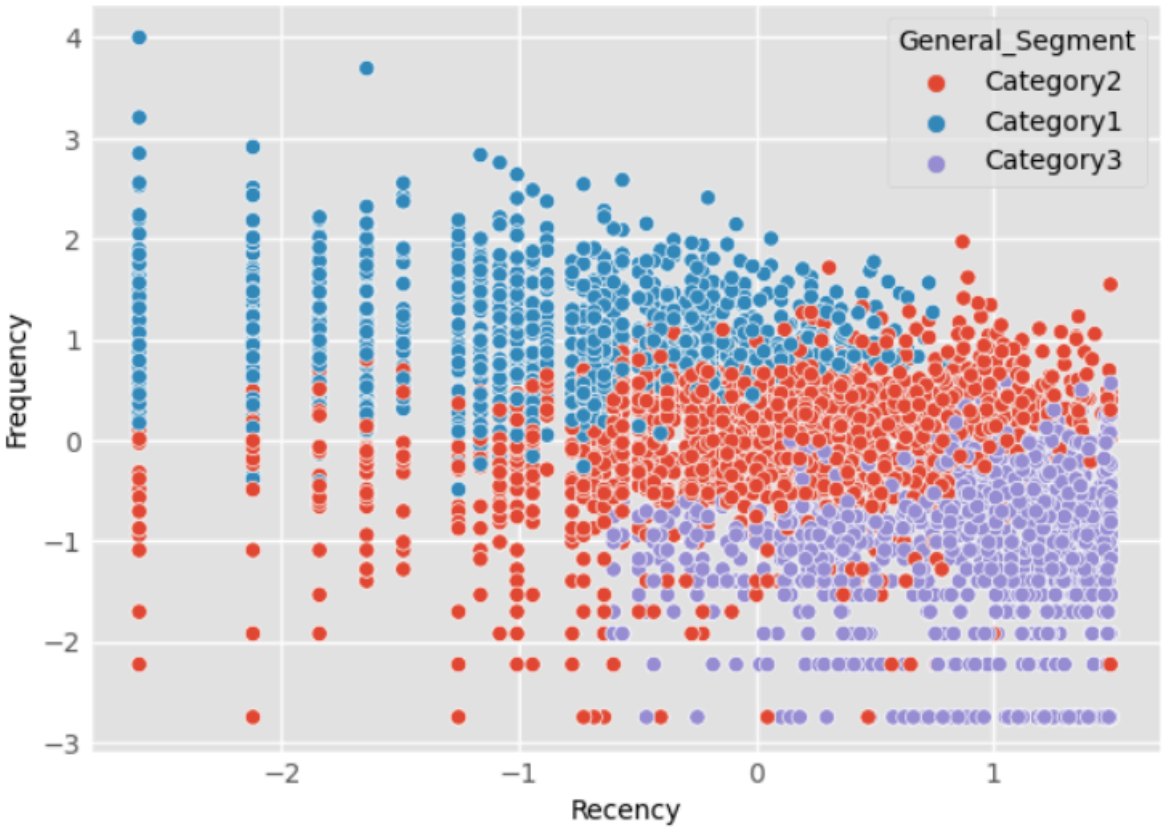
This is the unscaled general segmentation graph between requency and monetarvalue. Here the data is not scaled so it is clusterized at 1 point except that there are only ouliers. So it was not quite good for KMeans clustering.

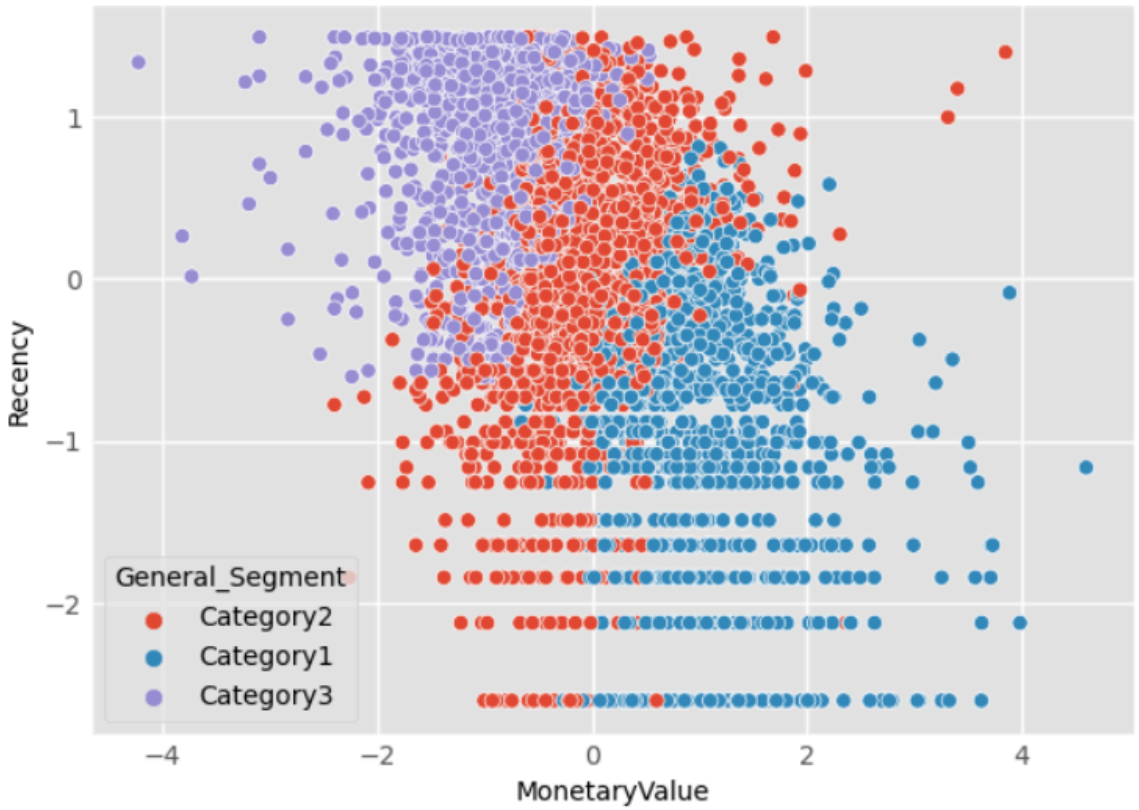
Similarly for other graphs


General_Segment
Category2
Category1
Category3
Frequency
Recency
8000
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300
350

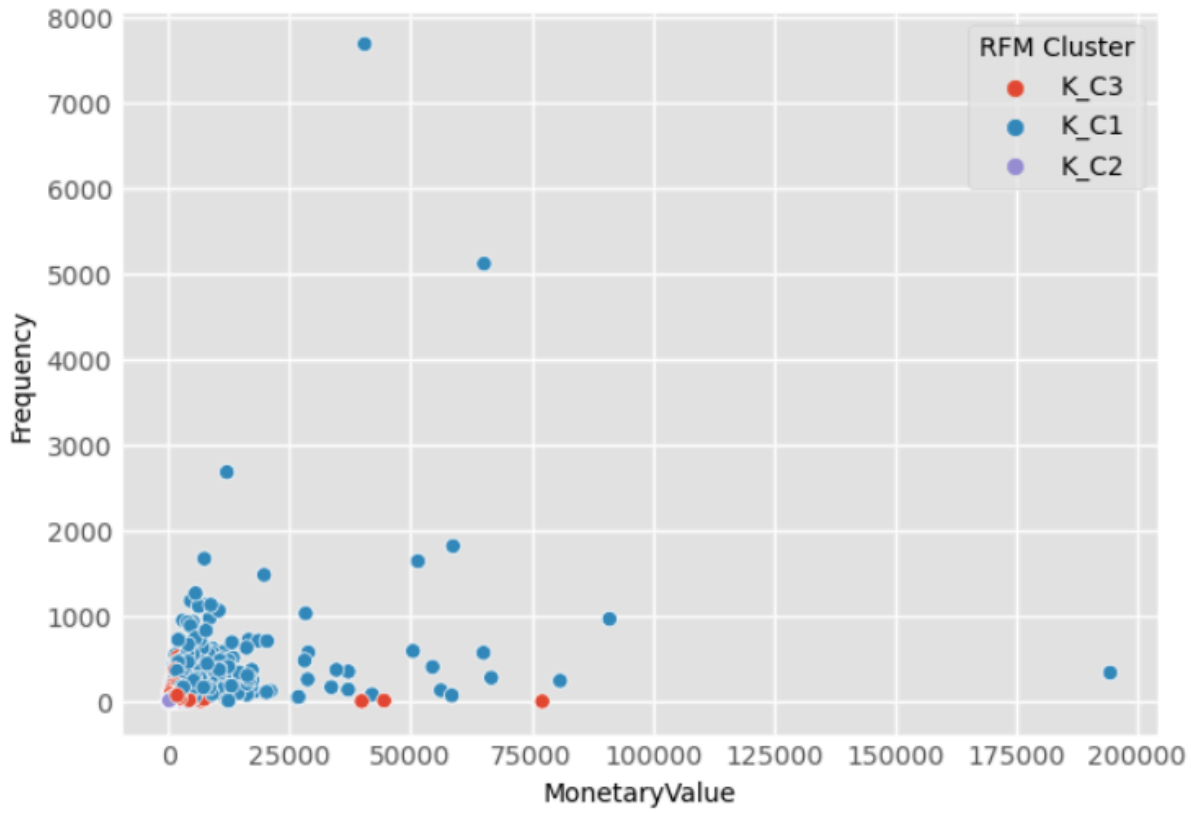

General_Segment
Category2
Category1
Category3
Recency
MonetaryValue
350
300
250
200
150
100
50
0
0
25000
50000
75000
100000
125000
150000
175000
200000

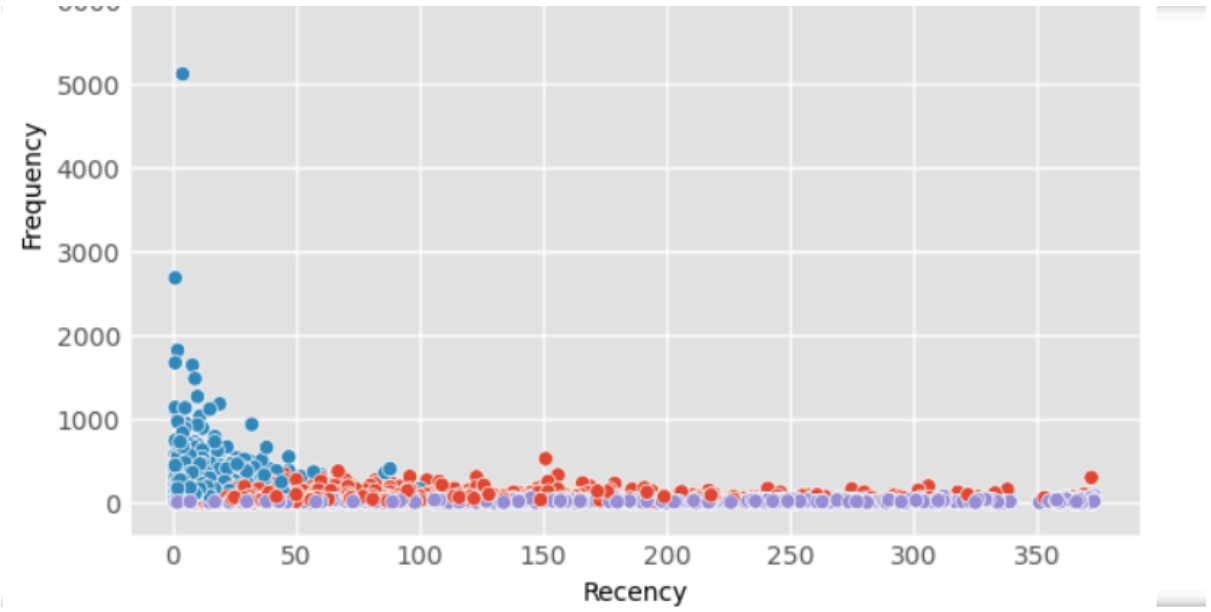
We scaled the data to apply KMeans clustering and for better visulisation.

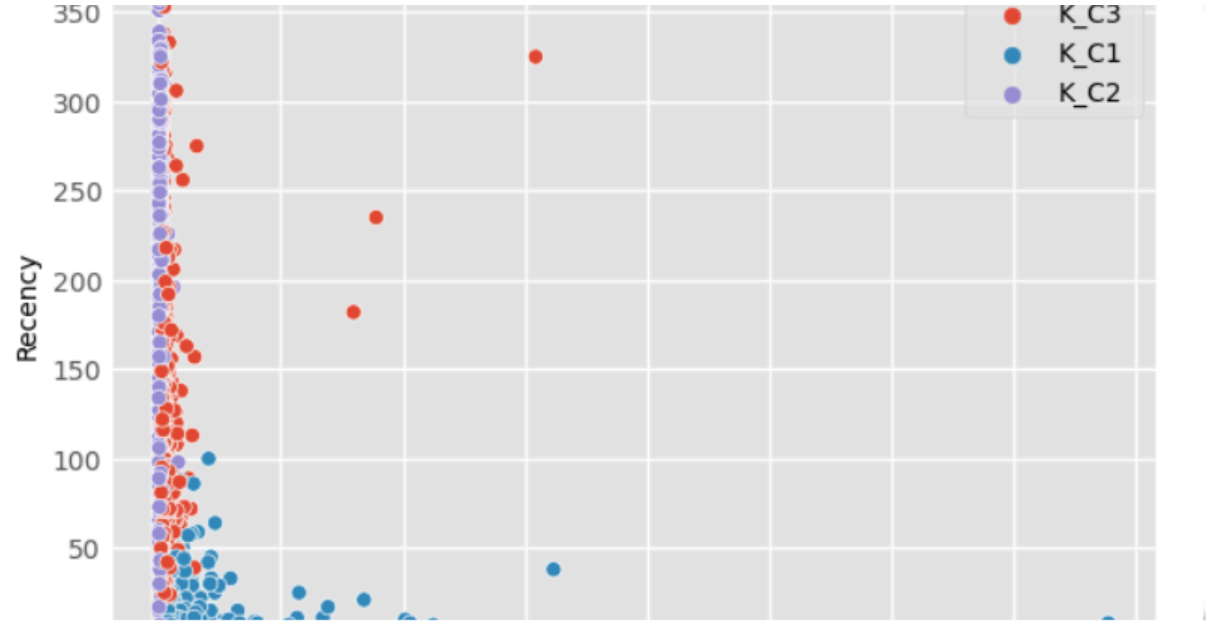




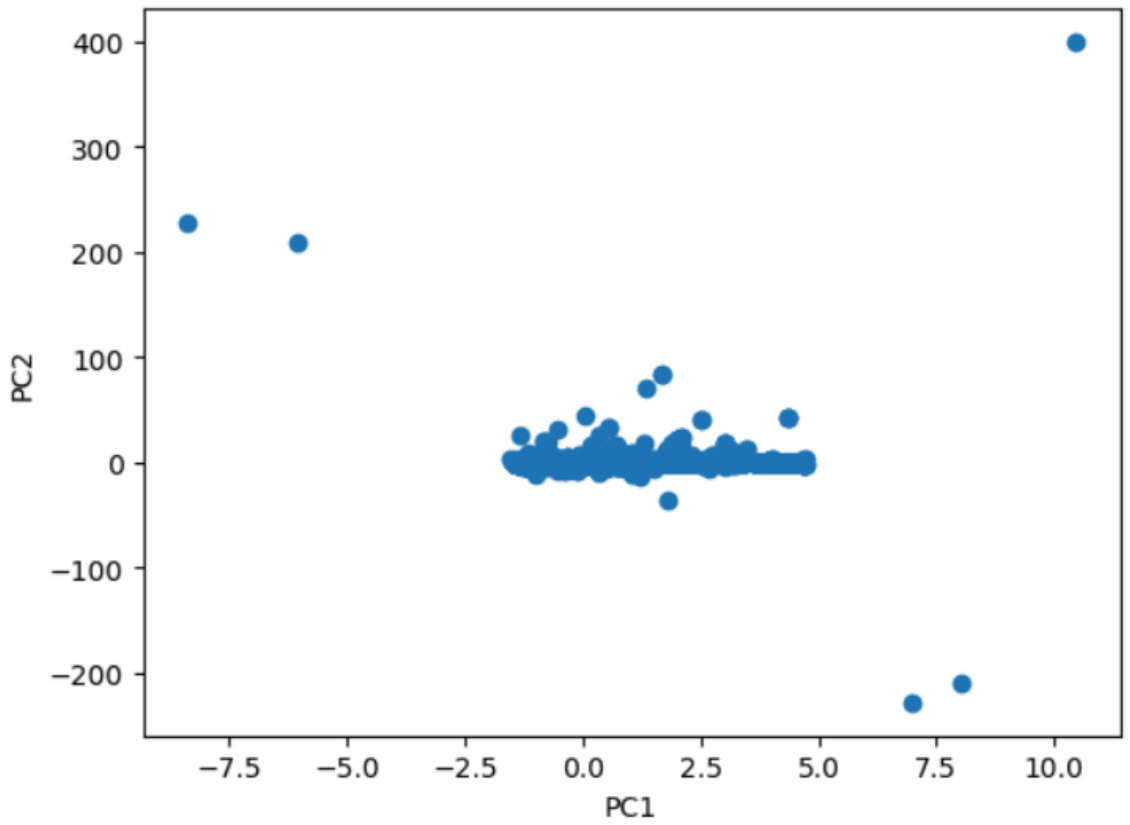
#Unscaled-KMeans







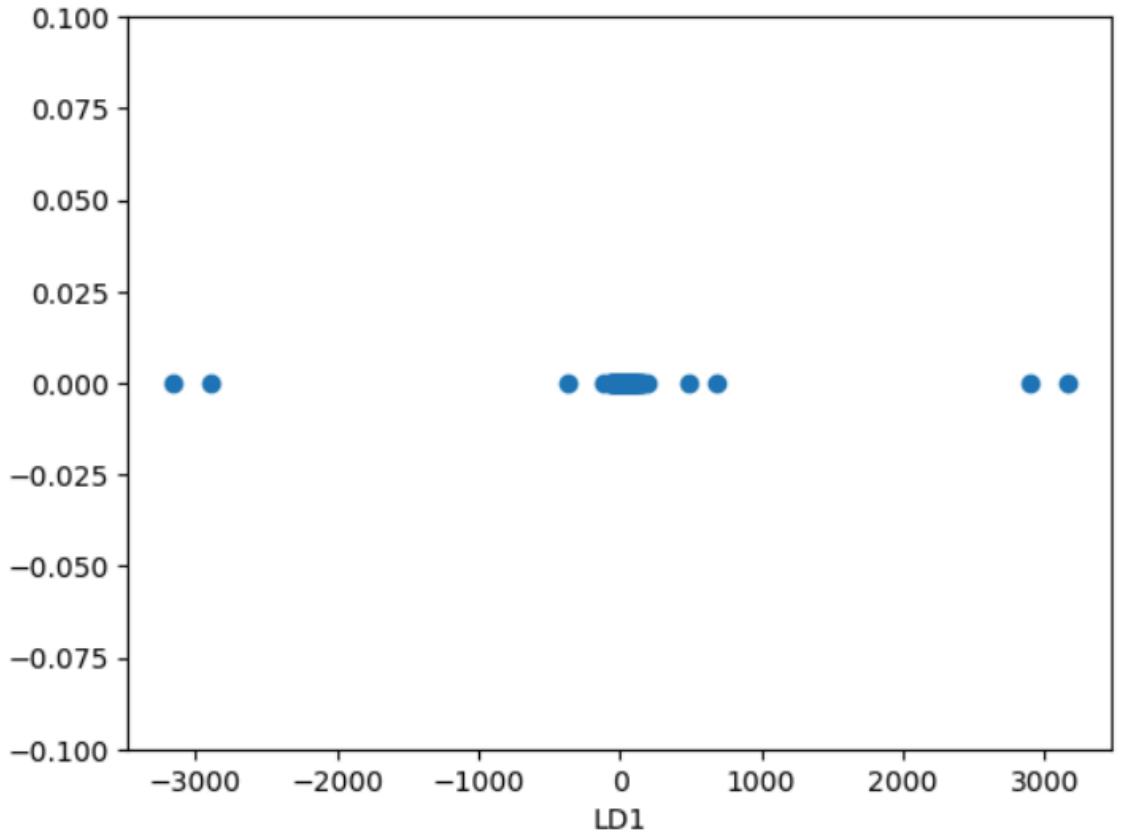
Now we apply PCA on it:



This it the graph we get using PCA.

We applied PCA and plotted the first two PCs on the scatter plot. The scatter plot helps us visualize the distribution of the data points in the reduced feature space, where each point represents a unique combination of the original features.

Now we apply LDA on it for further analysis



This is the graph we get for LDA

By applying LDA we obtain a set of liner combinations of the original features that can best separate the data into distinct groups based on the target variable.

We reduced the dimensionality of the dataset to one dimension using LDA, and obtained a single discriminant that maximally separates the transactions based on the InvoiceNo. The scatter plot help us visulize the relationship between transactions and the InvoiceNo variable. We may observe clusters of transctions that belong to the same InvoiceNo, which can indicate that these transactions are part of the same order or transaction.