Customer Segmentation
-RMF & K-Means clustering
REPORT

Submitted to:

Concerned Faculty
At
Great Learning

Submitted by:

Sashank Padmanabhuni
Aishwarya Nandini
Mukesh Pragada
Manvit CV
Jahanvi Virani
Sudarshan Jadhav

PGPDSE-FT Bangalore October 2023
Post Graduate Program in Data Science and Engineering

CONTENTS

Page no.

Introduction:

Our aim is to showcase the basics of data science (data cleaning, encoding, feature engineering, and model training), all while attempting to solve a problem that is common among businesses to segment the customer based on their purchase behaviour.

We will be treating this as a clustering problem, where we will attempt to create a model that clusters the customers into various bins based on their purchase behaviour.

The features of the Dataset:

'Invoice', 'StockCode', 'Description', 'Quantity', 'InvoiceDate', 'Price', 'Customer ID', 'Country'.

Problem Statement:

In the area of retail, understanding customer behavior is the Mainspring of success. Identifying distinct groups of customers, each with unique preferences and buying habit plays a key role in the success of the business. This is where the power of customer segmentation comes into play. Customer segmentation is a process of dividing all customers into distinct groups that share similar characteristics, such as demographics, interests, patterns, or location, and can help a business focus marketing efforts and resources on valuable, loyal customers to achieve business goals. This segmentation can be performed with customers' demographic, geographic, behavioural, psychological data and Clustering model.

Objective:

Take advantage of all of the feature variables available below, use it to Analyze and cluster the customers based on the purchase behavior.
This dataset contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011.The company mainly sells unique allocation gift-ware. Many customers of the company are wholesalers.

Data Dictionary:

1. Invoice No: Nominal. A 6-digit integral number uniquely assigned to each transaction. If the code begins with 'c', it in dictates a cancellation.
2. Stock Code: Nominal. A 5-digit integral number uniquely assigned to each distinct product (item).
3. Description: Nominal. The name of the product (item).
4. Quantity: Numeric. The quantities of each product (item) per transaction.
5. Invoice Date: Numeric. The date and time when a transaction was generated.
6. Price: Numeric. The unit price of the product per unit in sterling (£).
7. Customer ID: Nominal. A 5-digit integral number uniquely assigned to each customer.
8. Country: Nominal. The name of the country where a customer resides.

Exploratory Data Analysis:

Solution:

Firstly, after importing all the relevant libraries, we load the data set. Then, we performed EDA to extract and see patterns in the given data set.

Shape of the data:

The shape of the data is (1067371,8) that is the data consists of 1067371 rows and 8 columns.

Data types of the Data:

```
Invoice          object
StockCode        object
Description      object
Quantity          int64
InvoiceDate      object
Price           float64
Customer ID     float64
Country          object
dtype: object
```

First 5 records in the data:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-12-01 07:45:00	6.95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-12-01 07:45:00	2.10	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-12-01 07:45:00	1.25	13085.0	United Kingdom

Bottom 5 records in the data:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
1067366	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France
1067367	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France
1067368	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	France
1067369	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	France
1067370	581587	POST	POSTAGE	1	2011-12-09 12:50:00	18.00	12680.0	France

Information of the data:

```
#   Column        Non-Null Count    Dtype
---  ------        --------------    -----
0   Invoice       1067371 non-null  object
1   StockCode     1067371 non-null  object
2   Description   1062989 non-null  object
3   Quantity      1067371 non-null  int64
4   InvoiceDate   1067371 non-null  object
5   Price         1067371 non-null  float64
6   Customer ID   824364 non-null   float64
7   Country       1067371 non-null  object
```

5-point summary:

Numerical:

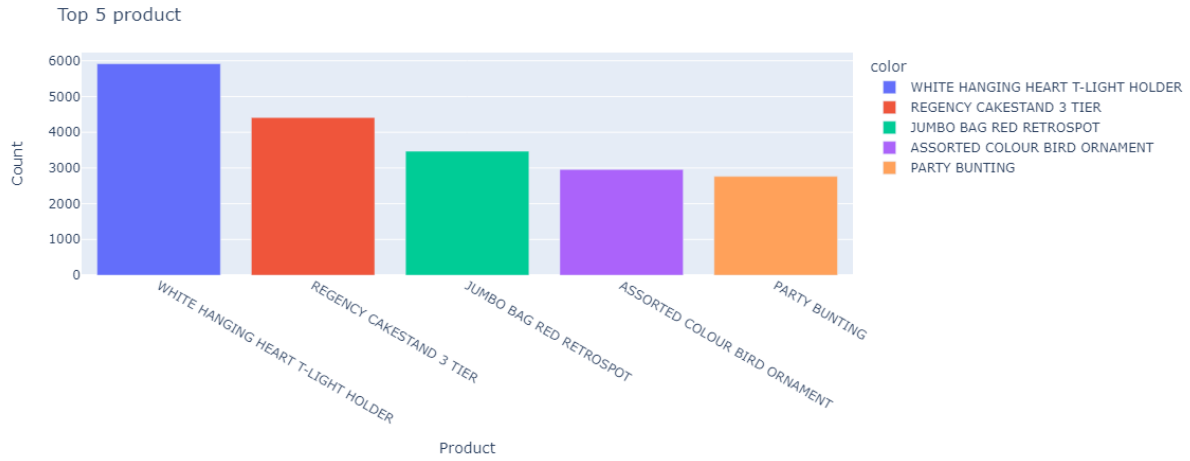
	count	mean	std	min	25%	50%	75%	max
Quantity	1067371.0	9.938898	172.705794	-80995.00	1.00	3.0	10.00	80995.0
Price	1067371.0	4.649388	123.553059	-53594.36	1.25	2.1	4.15	38970.0
Customer ID	824364.0	15324.638504	1697.464450	12346.00	13975.00	15255.0	16797.00	18287.0

Categorical:

	count	unique	top	freq
Invoice	1067371	53628	537434	1350
StockCode	1067371	5305	85123A	5829
Description	1062989	5698	WHITE HANGING HEART T-LIGHT HOLDER	5918
InvoiceDate	1067371	47635	2010-12-06 16:57:00	1350
Country	1067371	43	United Kingdom	981330

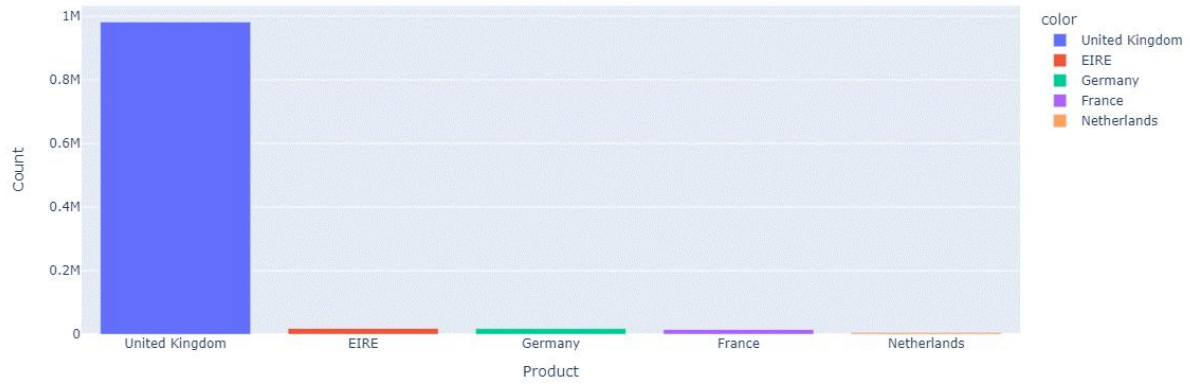
Top 5 Products sold:

```
Description
WHITE HANGING HEART T-LIGHT HOLDER    5918
REGENCY CAKESTAND 3 TIER              4412
JUMBO BAG RED RETROSPOT               3469
ASSORTED COLOUR BIRD ORNAMENT         2958
PARTY BUNTING                         2765
Name: count, dtype: int64
```



Top 5 Countries:

```
Country
United Kingdom    981330
EIRE               17866
Germany            17624
France             14330
Netherlands         5140
Name: count, dtype: int64
```



Data Preparation:

Treating the anomalies in the StockCode & Invoice column.

In Invoice column there are some invoices that starts with ‘C’ which represents cancelled, which doesn’t affect analysis in any way hence we are removing the cancelled invoices.

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
178	C489449	22087	PAPER BUNTING WHITE LACE	-12	2009-12-01 10:33:00	2.95	16321.0	Australia
179	C489449	85206A	CREAM FELT EASTER EGG BASKET	-6	2009-12-01 10:33:00	1.65	16321.0	Australia
180	C489449	21895	POTTING SHED SOW 'N' GROW SET	-4	2009-12-01 10:33:00	4.25	16321.0	Australia
181	C489449	21896	POTTING SHED TWINE	-6	2009-12-01 10:33:00	2.10	16321.0	Australia
182	C489449	22083	PAPER CHAIN KIT RETRO SPOT	-12	2009-12-01 10:33:00	2.95	16321.0	Australia

StockCode is a 5-digit numerical column, which are unique for each product, but in the data, we have records with StockCode greater than length 5 and alphabetical values in it, which represents sub-categories, postal charges, packaging charges, bank fee, portal fee and more.

Instead of removing these values we create a separate column for sub-categories and remove the additional fees and charges which are not actual product.

We used string manipulation technique and filtered all the characters which are greater than length 5 to subcat column.

Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country	Stock_subcat
489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-12-01 07:45:00	6.95	13085.0	United Kingdom	None
489434	79323P	PINK CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom	P
489434	79323W	WHITE CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom	W
489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-12-01 07:45:00	2.10	13085.0	United Kingdom	None
489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-12-01 07:45:00	1.25	13085.0	United Kingdom	None

There are 80 unique non numeric values present in StockCode.

```
array([None, 'P', 'W', 'C', 'B', 'F', 'L', 'S', 'A', 'N', 'POST', 'E',
       'J', 'D', 'G', 'LP', 'BL', 'K', 'H', 'M', '058', '068', 'DOT', 'U',
       'b', 'w', 'c', 'a', 'f', 'bl', 's', 'p', 'GR', 'R', 'V', '004',
       '076', 'C2', 'T', 'CHARGES', '003', 'I', 'O', 'Z', '01', '0001_80',
       '072', '0001_20', '044', '02', '0001_10', '0001_50', '066N', 'm',
       '0001_30', 'PADS', 'e', 'd', '0001_40', '0001_60', '0001_70',
       '0001_90', 'k', 'g', '069', '070', '075', 'j', 'l', '041', 'n',
       '037', 'BOY', 'GIRL', 'T2', 'J ', '2', '062', 'Y', 'NFEE'],
      dtype=object)
```

These 80 includes postal charges, platform fee, bank fee, package charges, delivery charges, gift cards and more.

After removing all the fees and charges which are not actual subcategory, we are down to 29 subcategories.

```
array([None, 'P', 'W', 'C', 'B', 'F', 'L', 'S', 'A', 'N', 'E', 'J', 'D',
       'G', 'LP', 'BL', 'K', 'H', 'M', 'U', 'GR', 'R', 'V', 'T', 'I', 'O',
       'Z', 'J ', 'Y'], dtype=object)
```

Removing -ve values in quantity

Quantity
-96
-240
-192
-50
-44

Removing records with price = 0

Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country	Stock_subcat
489659	21350	NaN	230	2009-12-01 17:39:00	0.0	NaN	United Kingdom	None
489781	84292	NaN	17	2009-12-02 11:45:00	0.0	NaN	United Kingdom	None
489825	22076	6 RIBBONS EMPIRE	12	2009-12-02 13:34:00	0.0	16126.0	United Kingdom	None
489882	35751	NaN	12	2009-12-02 16:22:00	0.0	NaN	United Kingdom	C
489898	79323	NaN	954	2009-12-03 09:40:00	0.0	NaN	United Kingdom	G

Null value treatment

```
Invoice          0.000000
StockCode        0.000000
Description      0.000000
Quantity         0.000000
InvoiceDate      0.000000
Price            0.000000
Customer ID     22.591397
Country          0.000000
Stock_subcat    87.954309
dtype: float64
```

In stock_subcat we have 87.9% of null values instead of deleting that column we can change the column to a binary column, if there is subcategory it is represented with 1 and if there is no subcategory it is represents with 0.

Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country	Sub_cat
489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-12-01 07:45:00	6.95	13085.0	United Kingdom	0
489434	79323	PINK CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom	1
489434	79323	WHITE CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom	1
489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-12-01 07:45:00	2.10	13085.0	United Kingdom	0
489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-12-01 07:45:00	1.25	13085.0	United Kingdom	0

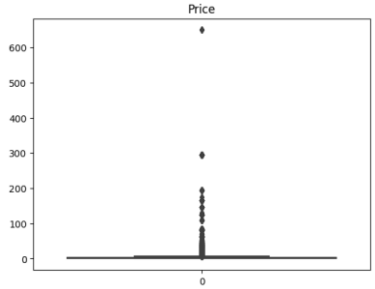
Now coming to Customer ID, we have 22.6% of missing values, the whole problem is to segment the customers into different clusters, we tried to map the customers based on the invoice number, but there is no match. Hence, we are deleting records with no customer ID, considering the problem statement and size of the dataset.

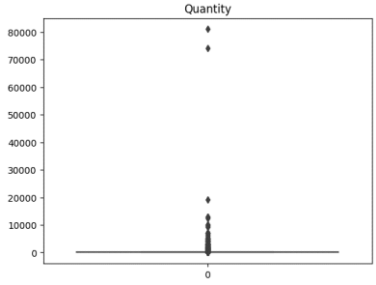
```
Invoice          0
StockCode        0
Description      0
Quantity         0
InvoiceDate      0
Price            0
Customer ID      0
Country          0
Sub_cat          0
dtype: int64
```

Datatype assigning and outlier treatment

Then we converted all the columns to appropriate datatypes and removed unwanted, invalid outliers from the data set.

```
df_order['Invoice']=df_order['Invoice'].astype(object)
df_order['StockCode']=df_order['StockCode'].astype(object)
df_order['Sub_cat']=df_order['Sub_cat'].astype(object)
df_order['Customer ID']=df_order['Customer ID'].astype(object)
```





Feature engineering

From InvoiceDate column we have derived various new columns such as, order_time, month, year, day, seasons, weekend, Hour.

order_time	month	year	day	seasons	weekend	Hour
07:45:00	December	2009	Tuesday	Winter	no	Morning
07:45:00	December	2009	Tuesday	Winter	no	Morning
07:45:00	December	2009	Tuesday	Winter	no	Morning
07:45:00	December	2009	Tuesday	Winter	no	Morning

After all treatment and feature engineering. We are now with 802632 rows and 17 columns.

```
#   Column        Non-Null Count   Dtype
--  ------        --------------   -----
0   Invoice       802632 non-null  object
1   StockCode     802632 non-null  object
2   Description   802632 non-null  object
3   Quantity      802632 non-null  int64
4   InvoiceDate   802632 non-null  datetime64[ns]
5   Price         802632 non-null  float64
6   Customer ID   802632 non-null  object
7   Country       802632 non-null  object
8   Sub_cat       802632 non-null  object
9   order_time    802632 non-null  object
10  month         802632 non-null  object
11  year          802632 non-null  object
12  day           802632 non-null  object
13  seasons       802632 non-null  object
14  weekend       802632 non-null  object
15  Hour          802632 non-null  object
16  total_amount  802632 non-null  float64
```

Analysis based on new features:

5-point summary:

Numerical:

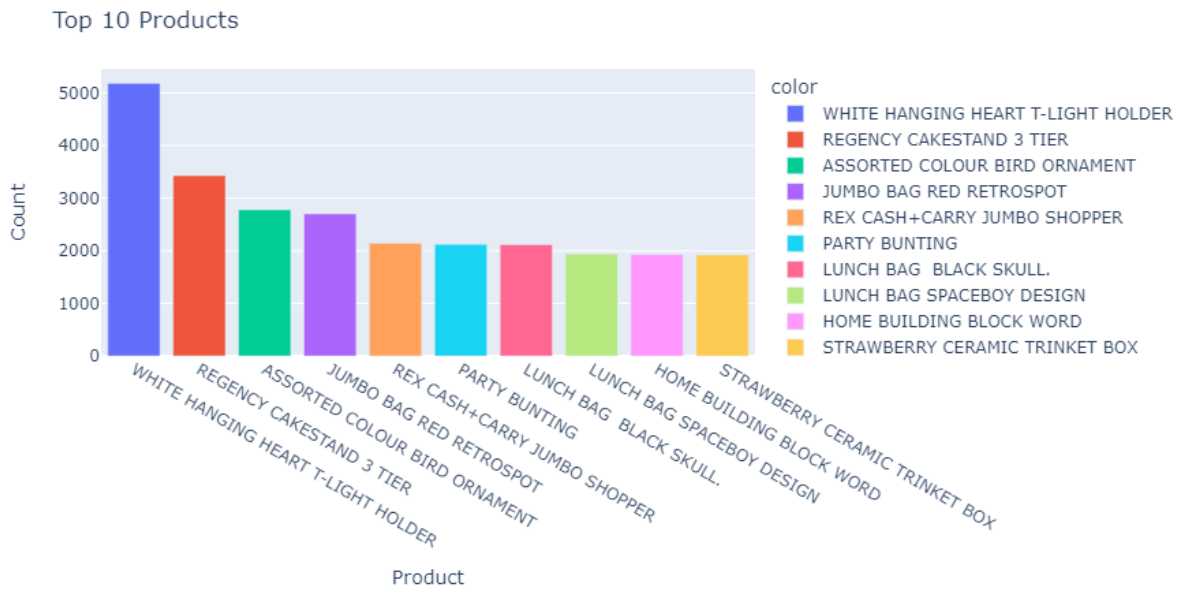
	count	mean	min	25%	50%	75%	max	std
Quantity	802632.0	13.319211	1.0	2.0	5.0	12.0	80995.0	143.868968
InvoiceDate	802632	2011-01-01 20:07:06.492838912	2009-12-01 00:00:00	2010-07-07 00:00:00	2010-12-03 00:00:00	2011-07-28 00:00:00	2011-12-09 00:00:00	NaN
Price	802632.0	2.929171	0.03	1.25	1.95	3.75	295.0	4.152337
total_amount	802632.0	21.673059	0.06	4.95	11.8	19.5	168469.6	218.284198

Categorical:

	count	unique	top	freq
Invoice	802632	36594	576339	541
StockCode	802632	3863	85099	6406
Description	802632	5269	WHITE HANGING HEART T-LIGHT HOLDER	5181
Customer ID	802632	5852	17841	12879
Country	802632	41	United Kingdom	724440
Sub_cat	802632	2	0	713989
order_time	802632	775	12:36:00	3418
month	802632	12	November	124425
year	802632	3	2010	401661
day	802632	7	Thursday	161056
seasons	802632	4	Autumn	297539
weekend	802632	2	no	666181
Hour	802632	3	Afternoon	553891

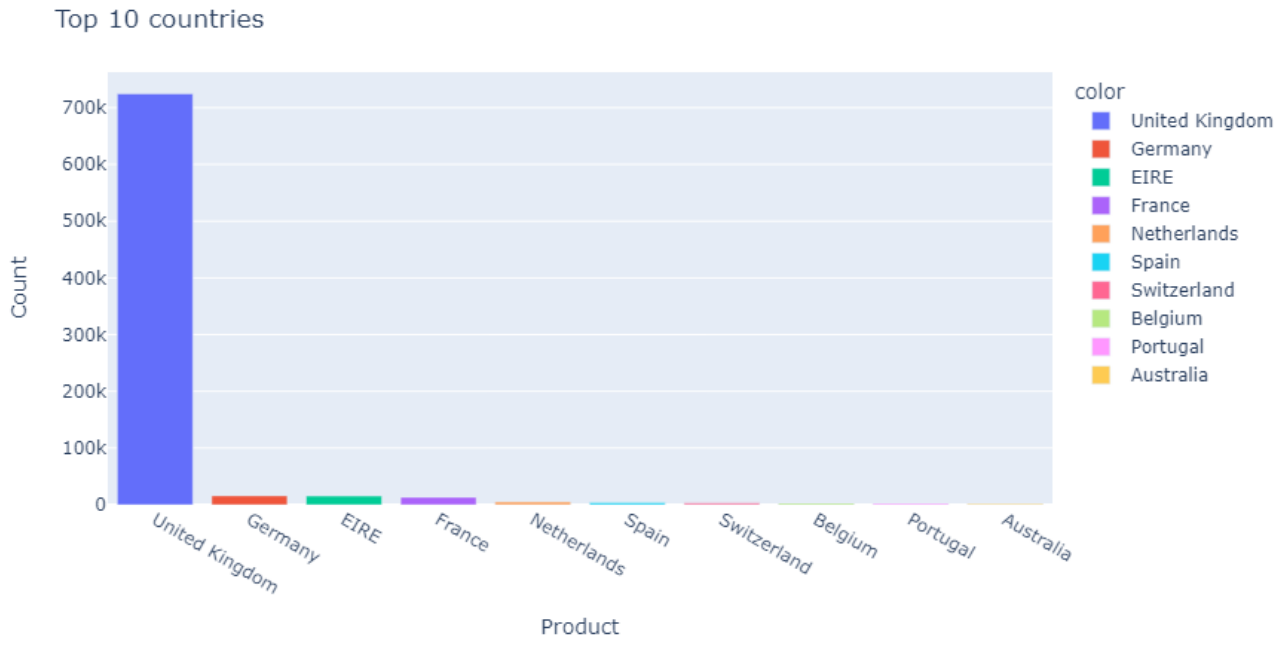
Top 10-products sold:

WHITE HANGING HEART T-LIGHT HOLDER	5181
REGENCY CAKESTAND 3 TIER	3428
ASSORTED COLOUR BIRD ORNAMENT	2777
JUMBO BAG RED RETROSPOT	2702
REX CASH+CARRY JUMBO SHOPPER	2141
PARTY BUNTING	2121
LUNCH BAG BLACK SKULL.	2117
LUNCH BAG SPACEBOY DESIGN	1941
HOME BUILDING BLOCK WORD	1929
STRAWBERRY CERAMIC TRINKET BOX	1922



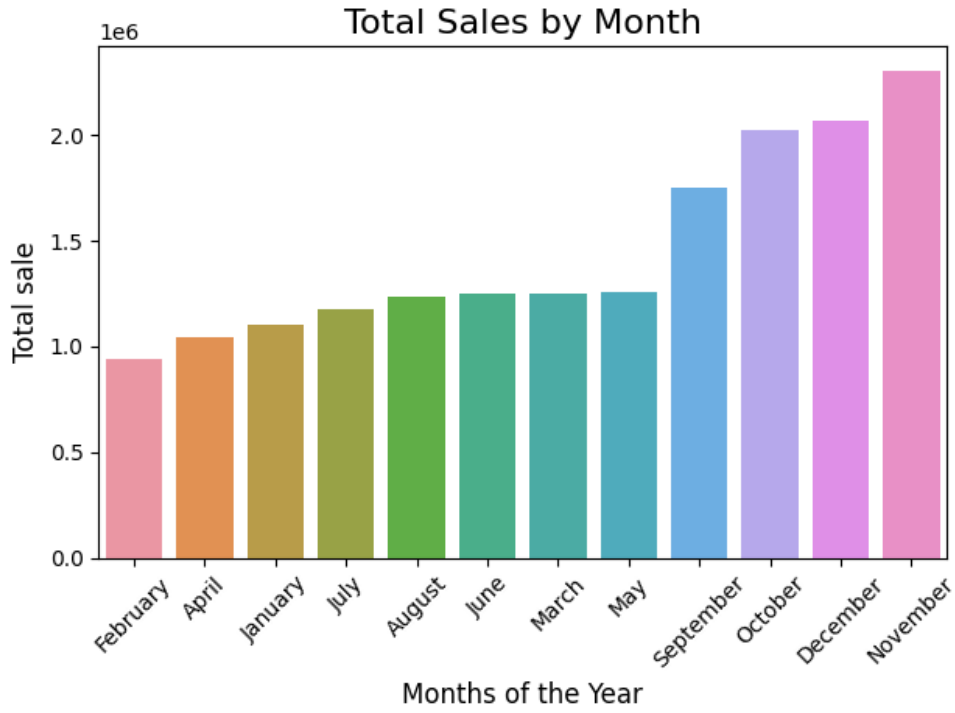
Top-10 Countries:

Country	
United Kingdom	724440
Germany	16034
EIRE	15524
France	13322
Netherlands	4983
Spain	3617
Switzerland	2956
Belgium	2923
Portugal	2381
Australia	1806



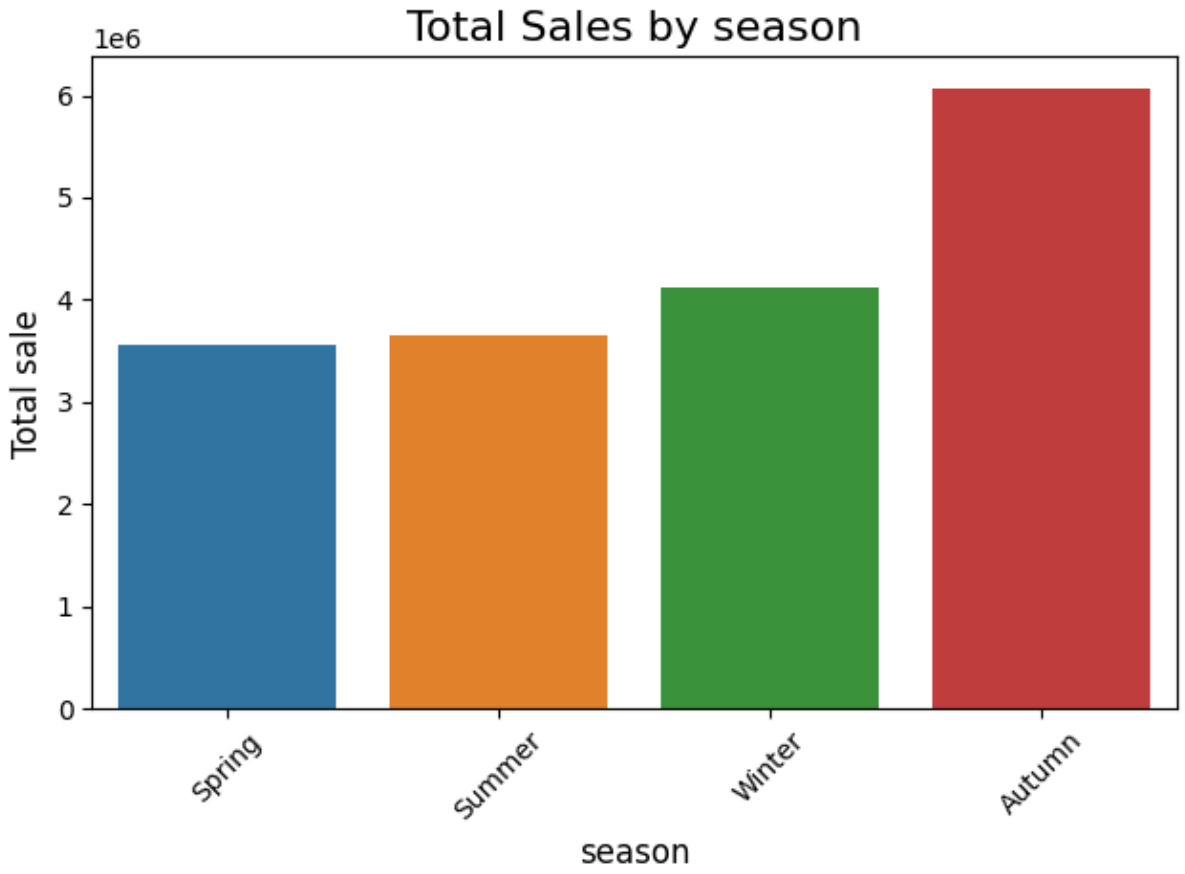
Monthly sales by total amount:

month	total_amount
February	943095.700
April	1043020.680
January	1099256.010
July	1176120.430
August	1233386.460
June	1246978.235
March	1253140.040
May	1255368.440
September	1749045.870
October	2020767.640
December	2071311.120
November	2304000.180



Season wise sales:

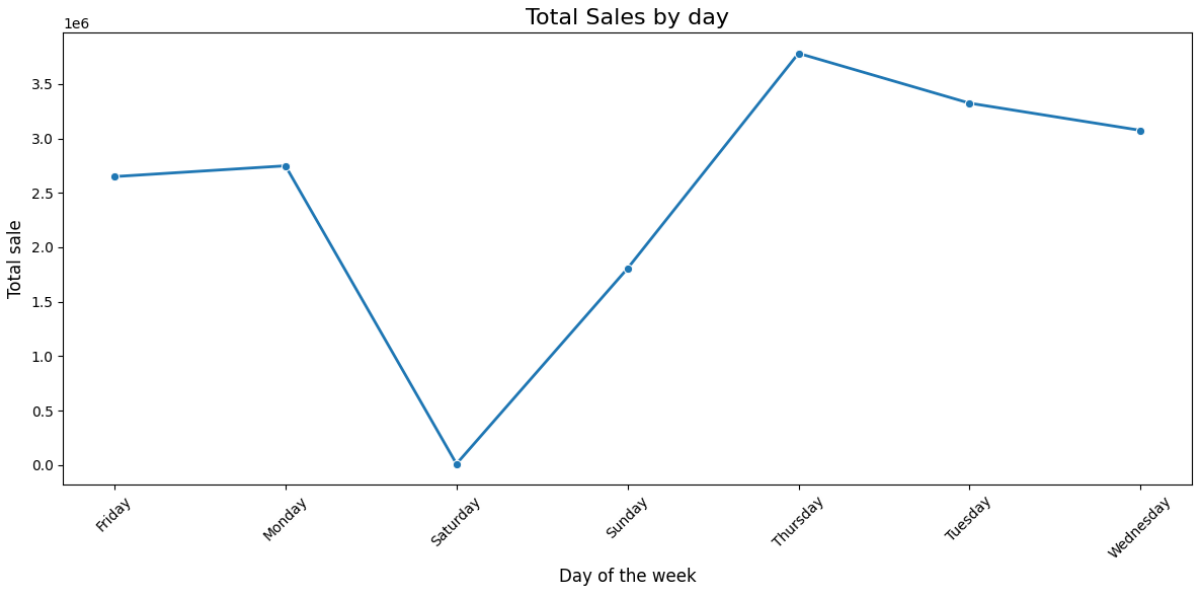
	seasons	total_amount
1	Spring	3551529.160
2	Summer	3656485.125
3	Winter	4113662.830
0	Autumn	6073813.690



Top 5-products quantity wise:

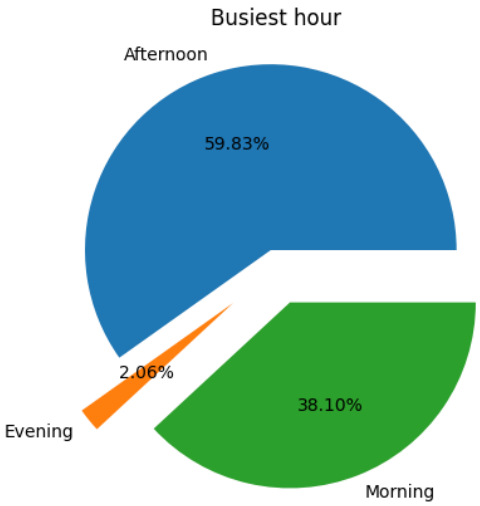
Description	Quantity
WORLD WAR 2 GLIDERS ASSTD DESIGNS	109169
WHITE HANGING HEART T-LIGHT HOLDER	93640
PAPER CRAFT , LITTLE BIRDIE	80995
ASSORTED COLOUR BIRD ORNAMENT	79913
MEDIUM CERAMIC TOP STORAGE JAR	77916

Day wise sales:



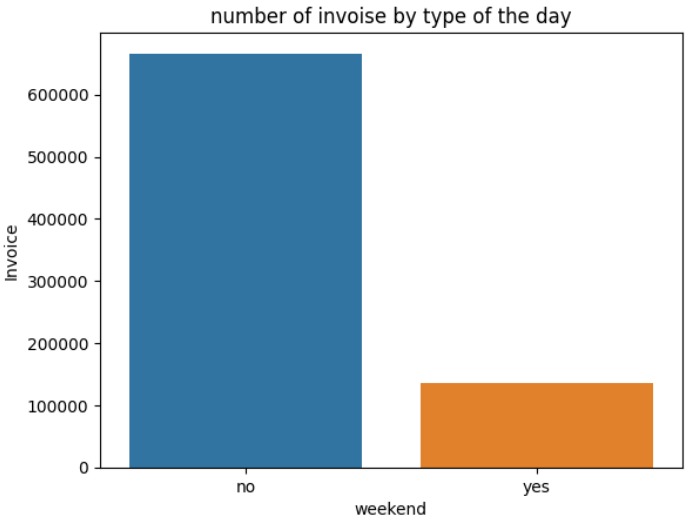
Busiest hour in the day:

Hour	total_amount
Afternoon	1.040824e+07
Evening	3.587792e+05
Morning	6.628471e+06

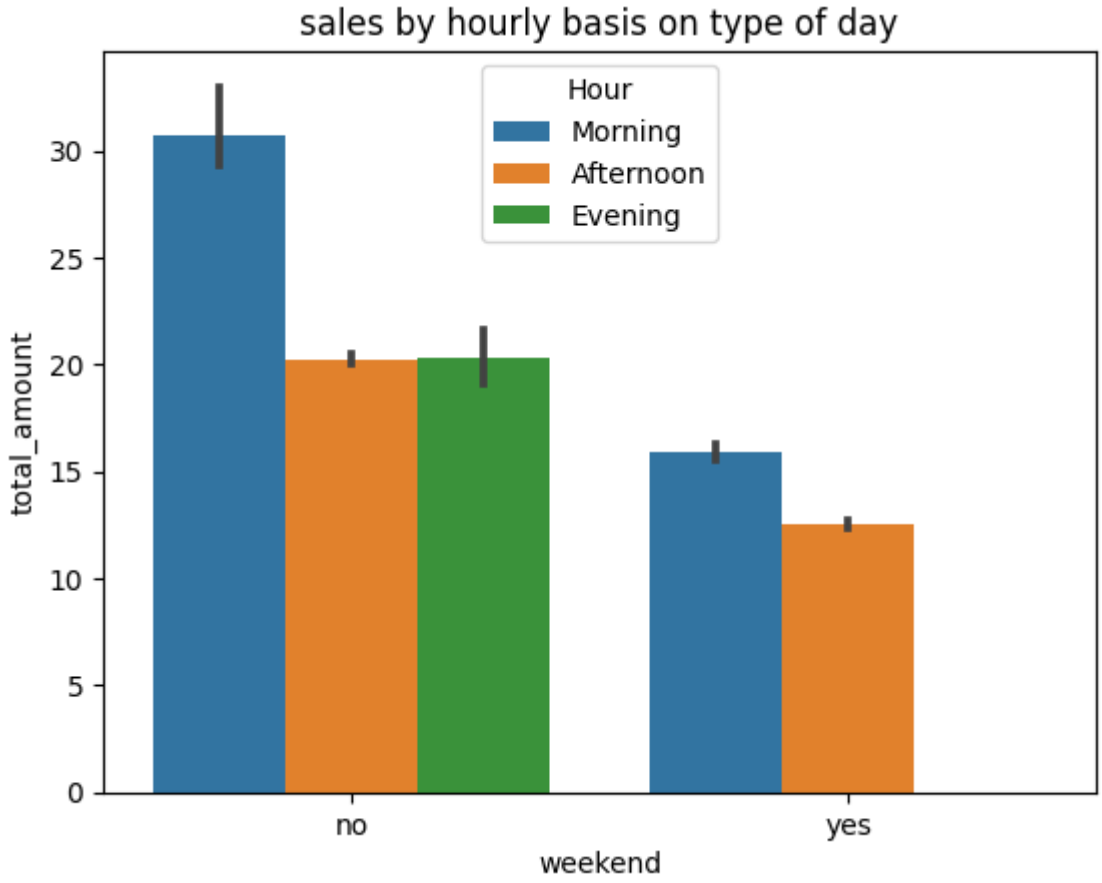


Sales by type of the day:

weekend	Invoice
no	666181
yes	136451



sales by hourly basis on type of day



RFM (Recency, Frequency, Monetary) Analysis:

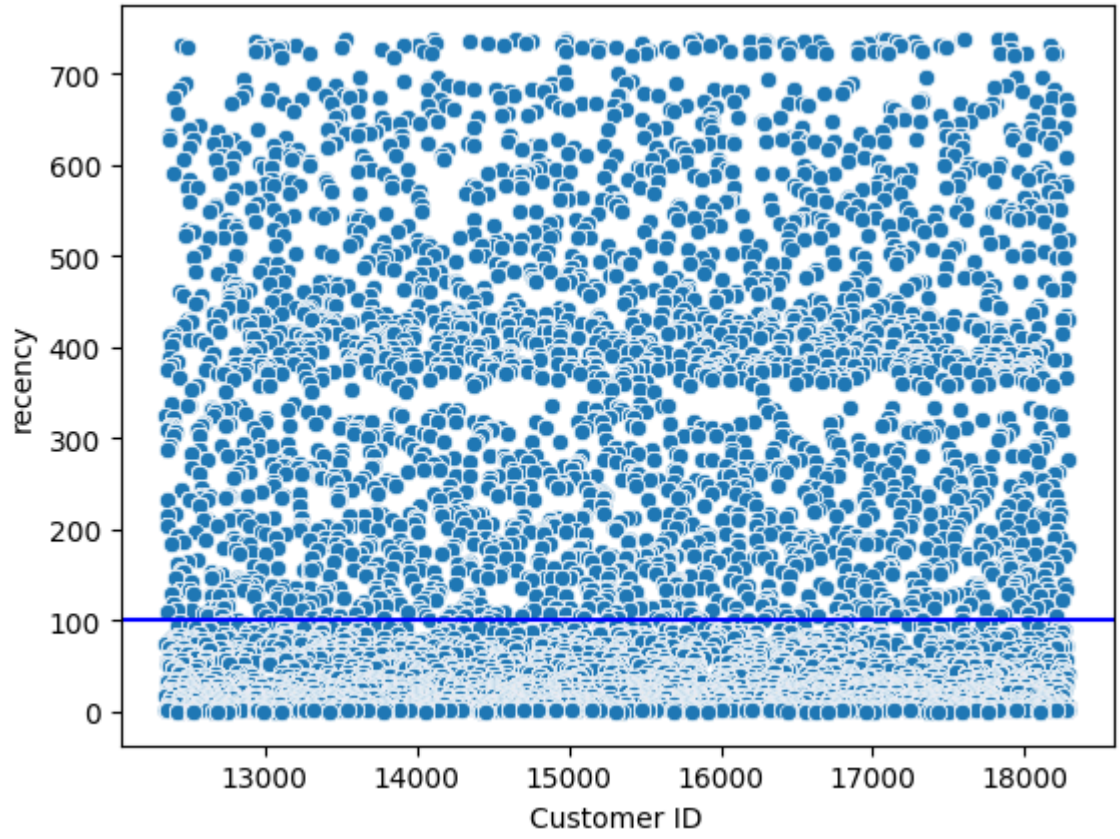
RFM analysis, or RFM stands for recency, frequency, and monetary value, is a marketing technique that uses data about customer behavior to segment customers and prioritize them. It's commonly used in direct marketing and database marketing, and has received particular attention in the retail and professional services industries.

- **Recency: The freshness of the customer activity, be it purchases.**
- **Frequency: The frequency of the customer transactions.**
- **Monetary: The intention of customer to spend or purchasing power of customers.**

Recency: calculating the days since last purchase date for each customers taking last billed invoice date as reference. hence the data is between 2009 and 2011, we consider the date as the last billing date in the entire dataset as reference date.

Customer ID	recency
14654	738.0
17592	738.0
13526	738.0
17056	738.0
17087	737.0

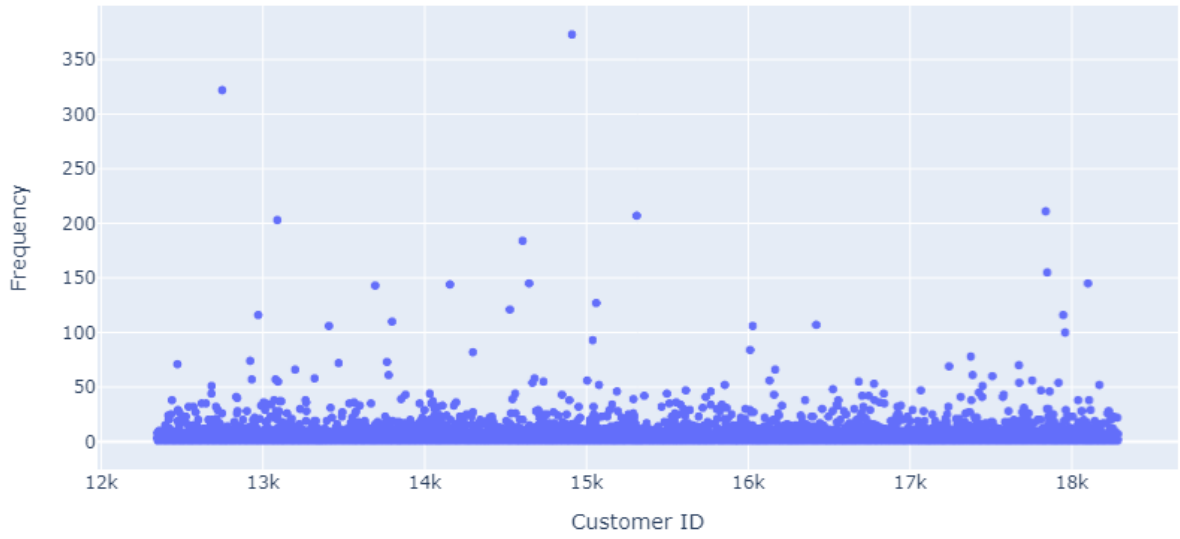
Distribution of Recency:



Frequency: calculating the frequency of purchase each customer made.

Customer ID	Frequency
14911	373
12748	322
17841	211
15311	207
13089	203

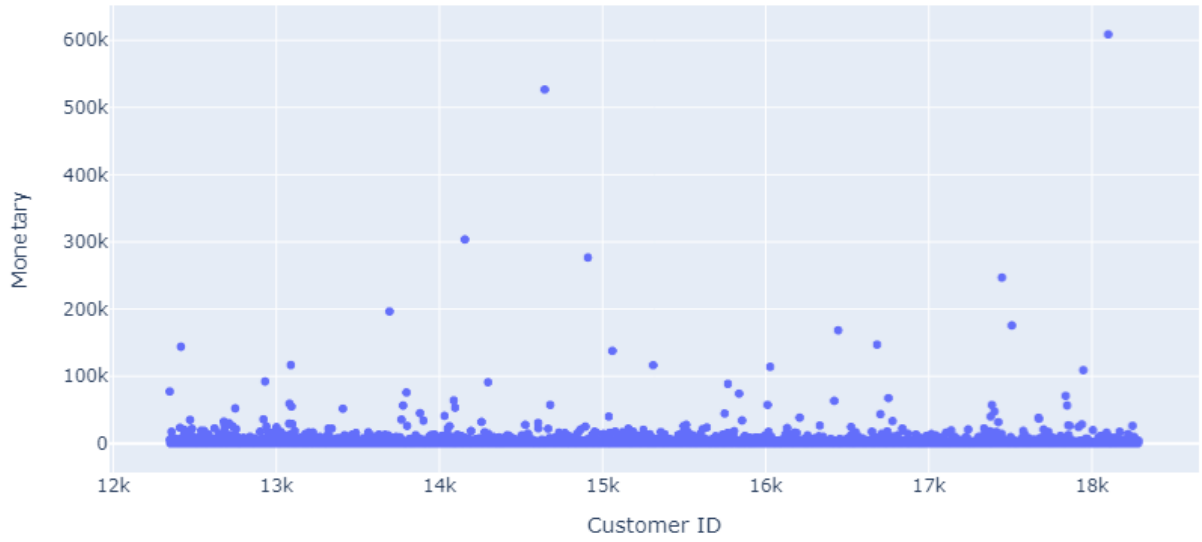
Frequency distribution:



Monetary: calculating the total amount spend by each customer.

Customer ID	Monetary
18102	608821.65
14646	526751.52
14156	303578.63
14911	276654.61
17450	246973.09

Monetary distribution:



RFM Table formation:

Combining all the recency, frequency and monetary values into a single table

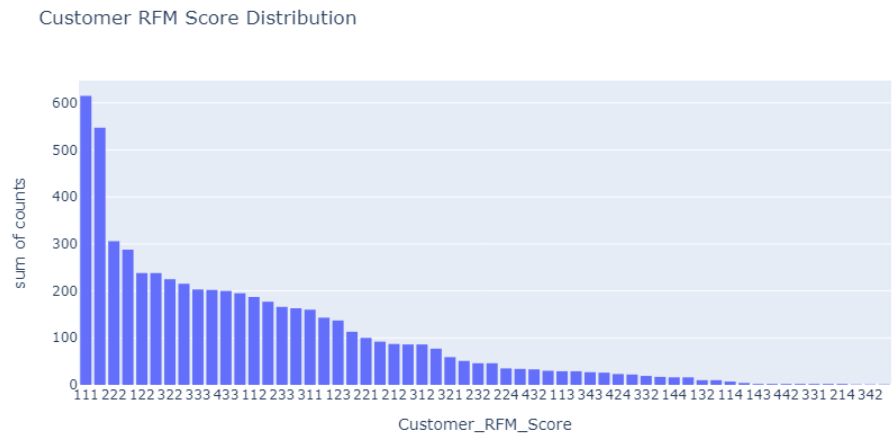
Customer ID	recency	Frequency	Monetary
12346	325.0	3	77352.96
12347	2.0	8	5633.32
12348	75.0	5	1658.40
12349	18.0	3	3678.69
12350	310.0	1	294.40

Calculating scores for all recency, frequency and monetary values by splitting them into 4 bins and combining all the scores to form RFM_Score.

	count	mean	std	min	25%	50%	75%	max
recency	5852.0	199.732399	208.523287	0.00	25.000	95.000	379.0000	738.00
Frequency	5852.0	6.253247	12.749286	1.00	1.000	3.000	7.0000	373.00
Monetary	5852.0	2972.571908	14597.005578	2.95	344.975	880.375	2288.7725	608821.65

Customer ID	recency	Frequency	Monetary	Recency_Score	Frequency_Score	Monetary_Score	RFM_Score
12346	325.0	3	77352.96	2	2	4	224
12347	2.0	8	5633.32	4	3	4	434
12348	75.0	5	1658.40	3	3	3	333
12349	18.0	3	3678.69	4	2	4	424
12350	310.0	1	294.40	2	1	1	211

Distribution of RFM_SCORE:

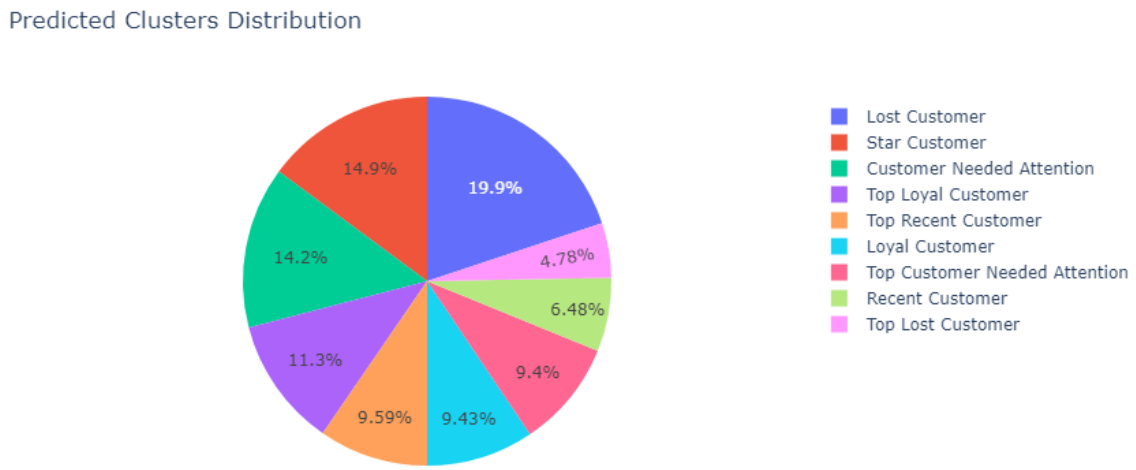


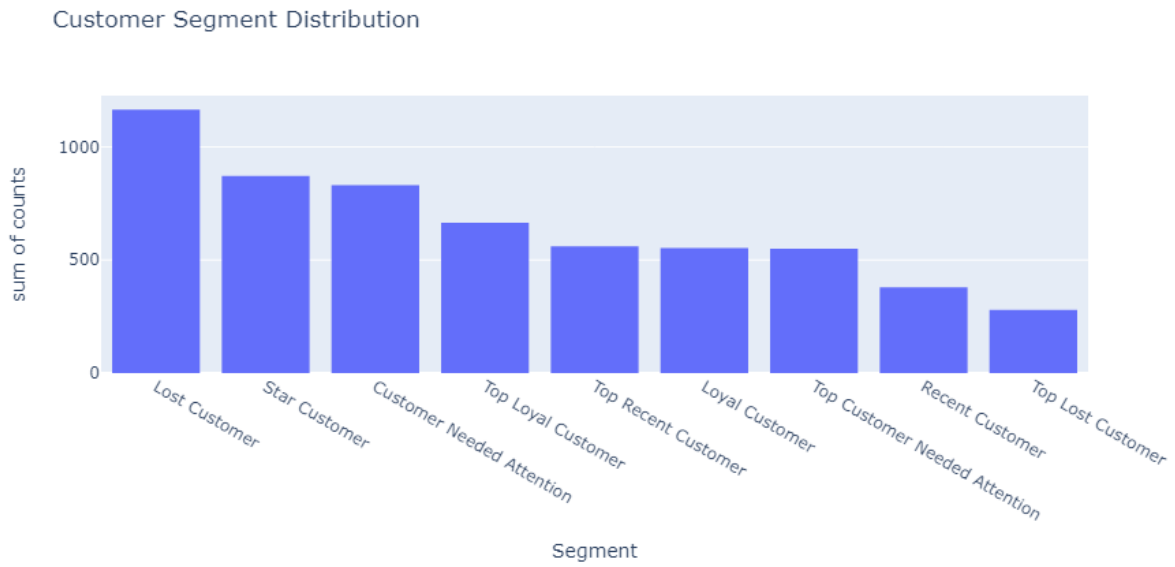
Now based on the RFM_Score we segment the customers into 9 different segments.

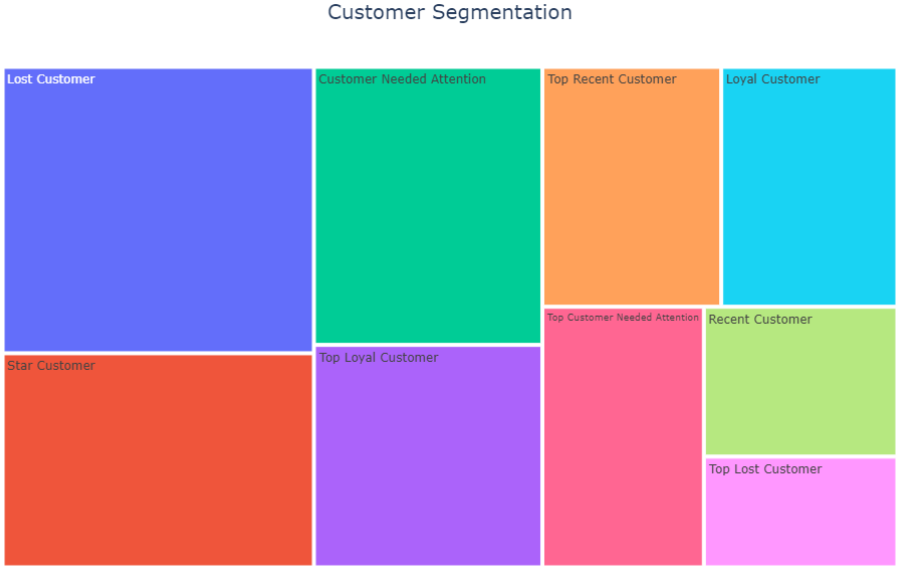
	Customer_Segment	RFM
0	Star Customer	(2|3|4)-(4)-(4)
1	Top Loyal Customer	(3)-(1|2|3|4)-(3|4)
2	Loyal Customer	(3)-(1|2|3|4)-(1|2)
3	Top Recent Customer	(4)-(1|2|3|4)-(3|4)
4	Recent Customer	(4)-(1|2|3|4)-(1|2)
5	Top Customer Needed Attention	(2|3)-(1|2|3|4)-(3|4)
6	Customer Needed Attention	(2|3)-(1|2|3|4)-(1|2)
7	Top Lost Customer	(1)-(1|2|3|4)-(3|4)
8	Lost Customer	(1)-(1|2|3|4)-(1|2)

Customer ID	recency	Frequency	Monetary	Recency_Score	Frequency_Score	Monetary_Score	RFM_Score	Segment
12346	325.0	3	77352.96	2	2	4	224	Top Customer Needed Attention
12347	2.0	8	5633.32	4	3	4	434	Top Recent Customer
12348	75.0	5	1658.40	3	3	3	333	Top Loyal Customer
12349	18.0	3	3678.69	4	2	4	424	Top Recent Customer
12350	310.0	1	294.40	2	1	1	211	Customer Needed Attention

Distribution of segments



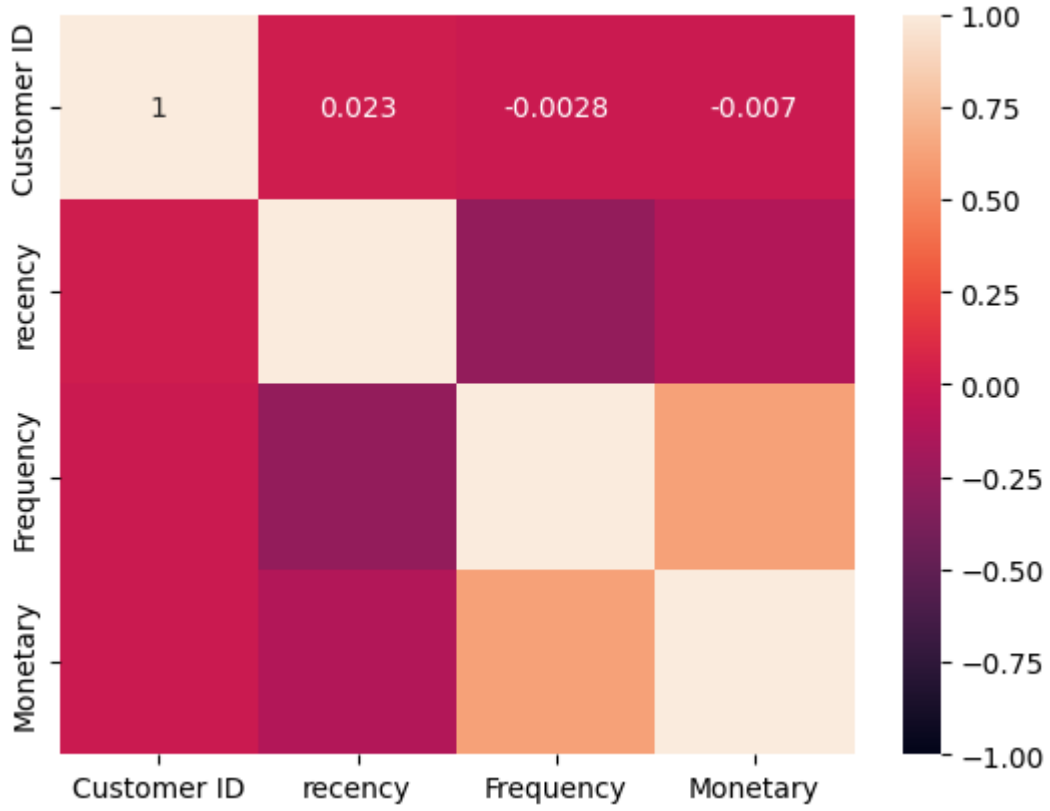




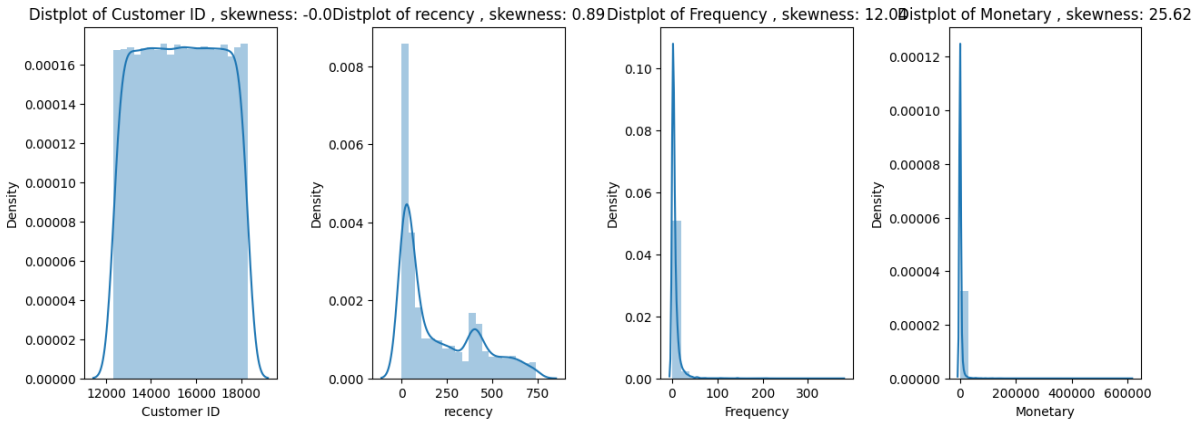
Segment	count
Lost Customer	1165
Star Customer	871
Customer Needed Attention	830
Top Loyal Customer	664
Top Recent Customer	561
Loyal Customer	552
Top Customer Needed Attention	550
Recent Customer	379
Top Lost Customer	280

Before creating a base model. We'll treat the data.

	Customer ID	recency	Frequency	Monetary
Customer ID	1.000000	0.023182	-0.002791	-0.006969
recency	0.023182	1.000000	-0.258582	-0.124497
Frequency	-0.002791	-0.258582	1.000000	0.621731
Monetary	-0.006969	-0.124497	0.621731	1.000000

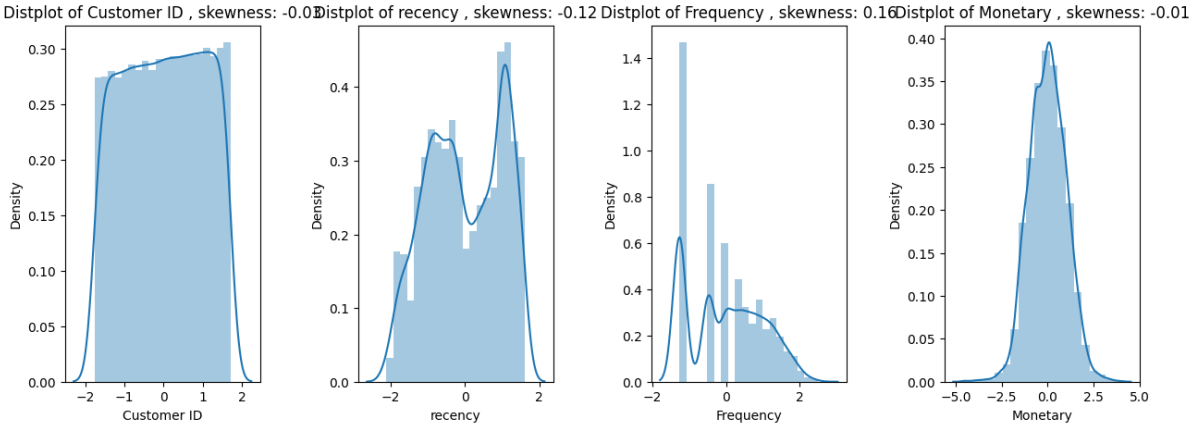


There is almost no relation between the data.



From the above we can see that there is skewness in the RFM values.

We, treated it using power-transform technique.



Customer ID	-0.034153
recency	-0.120408
Frequency	0.164823
Monetary	-0.008152

Scaling:

We used standard scaler to scale the data

	recency	Frequency	Monetary
0	0.862252	-0.011139	2.889383
1	-1.761405	0.966796	1.289818
2	-0.213970	0.531303	0.458685
3	-0.995211	-0.011139	1.006616
4	0.822562	-1.264585	-0.816564

Base-model:

We've considered k-means clustering as base model, for the clustering with default hyper parameters.

```
# base model
kmeans = KMeans().fit(rfm_scaled)
kmeans.fit_predict(rfm_scaled)
kmeans
```

✓ 0.0s

```
KMeans()
```

```
print(kmeans.inertia_,
silhouette_score(rfm_scaled, kmeans.labels_))
```

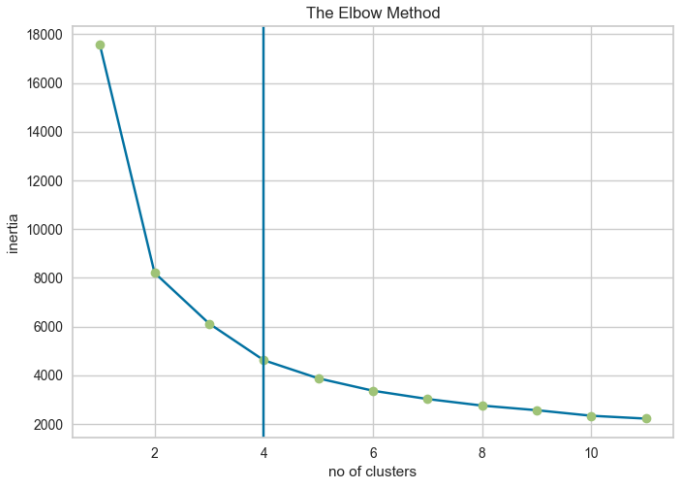
✓ 0.7s

```
2750.0169926304834 0.2999170997919618
```

Model Evaluation

Then we used various model evaluation parameters such as elbow plot, Silhouette score and Silhouette visualizer to find the optimal k value.

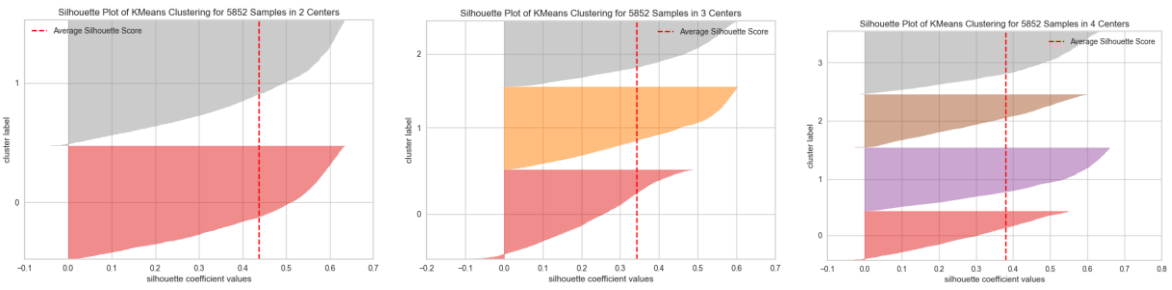
Elbow plot:



Silhouette score:

```
Silhouette Score for 2 clusters: 0.43756062318670497
Silhouette Score for 3 clusters: 0.34321984465356153
Silhouette Score for 4 clusters: 0.3800154996573313
Silhouette Score for 5 clusters: 0.3451571813274592
Silhouette Score for 6 clusters: 0.3351222874725948
Silhouette Score for 7 clusters: 0.2956791550973372
Silhouette Score for 8 clusters: 0.29509067539320777
Silhouette Score for 9 clusters: 0.29870020973087713
Silhouette Score for 10 clusters: 0.2901739208348717
Silhouette Score for 11 clusters: 0.28414028118346196
```

Silhouette visualizer:



Based on the above scores clusters with k=2 have better values compared to other. Even though splitting or clustering the entire data into just 2 big and vast clusters doesn't give much insights.

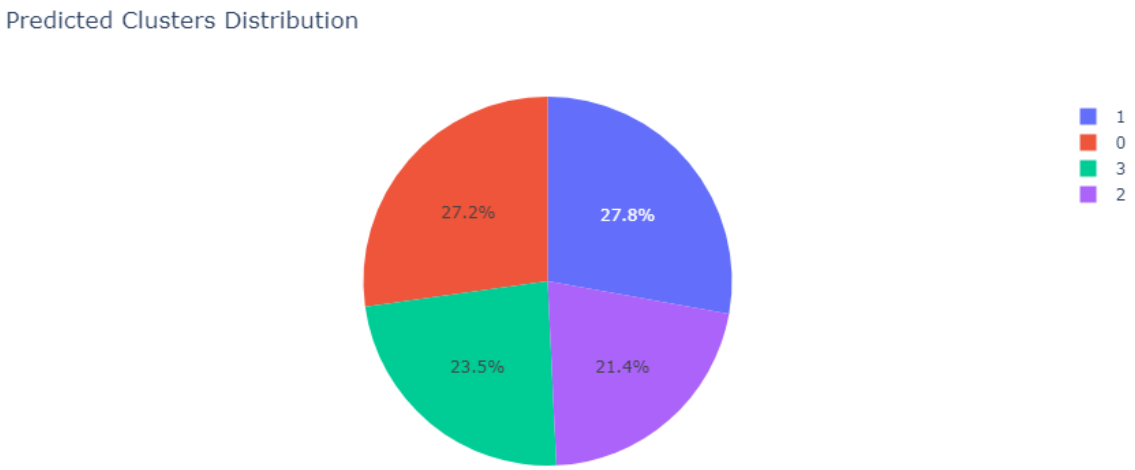
Final model:

Hence, we conclude that the next best is K=4 is a good number of clusters and built the final model on k=4.

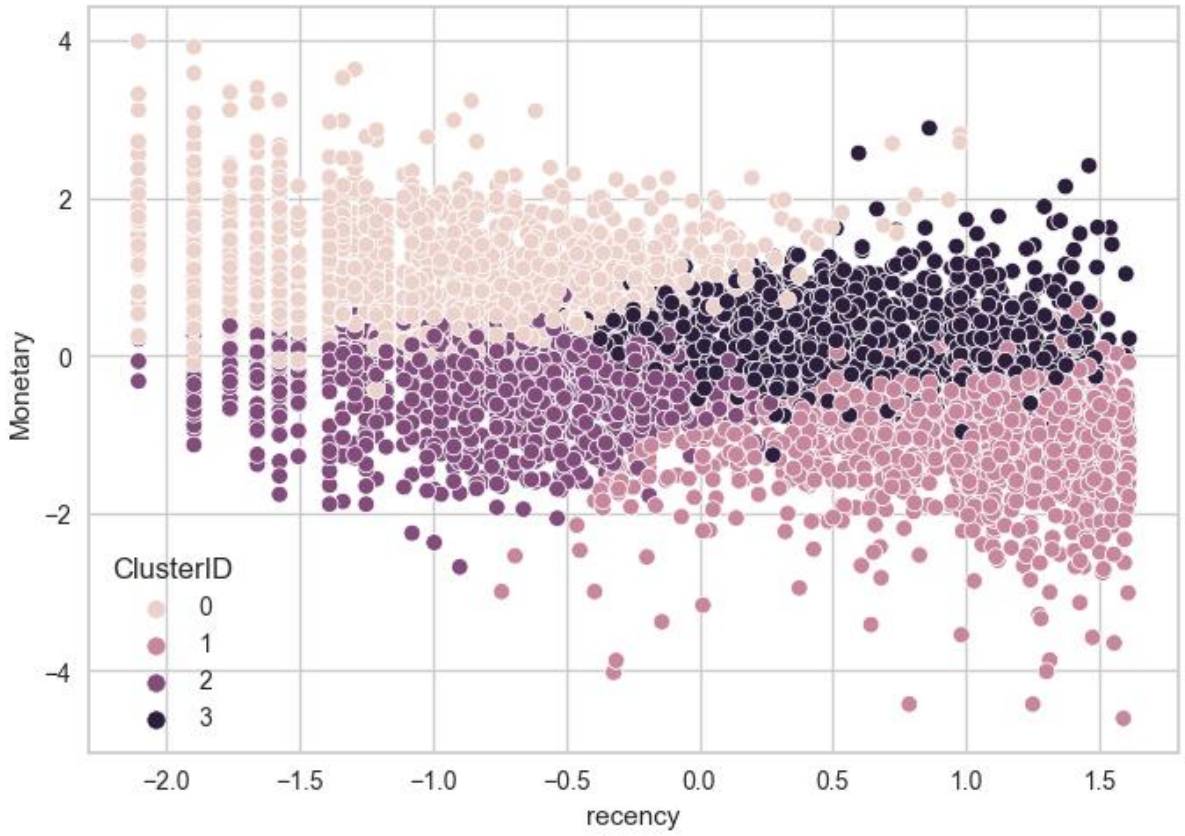
```
KMeans
KMeans(n_clusters=4, random_state=1)
```

	recency	Frequency	Monetary	ClusterID
0	0.862252	-0.011139	2.889383	3
1	-1.761405	0.966796	1.289818	0
2	-0.213970	0.531303	0.458685	3
3	-0.995211	-0.011139	1.006616	0
4	0.822562	-1.264585	-0.816564	1

Spread of clusters formed:



Visualizing the clusters formed between recency and monetary:



We can see that people who spend more money has also made the purchase very recently.

We assign the clusters to the original RFM table

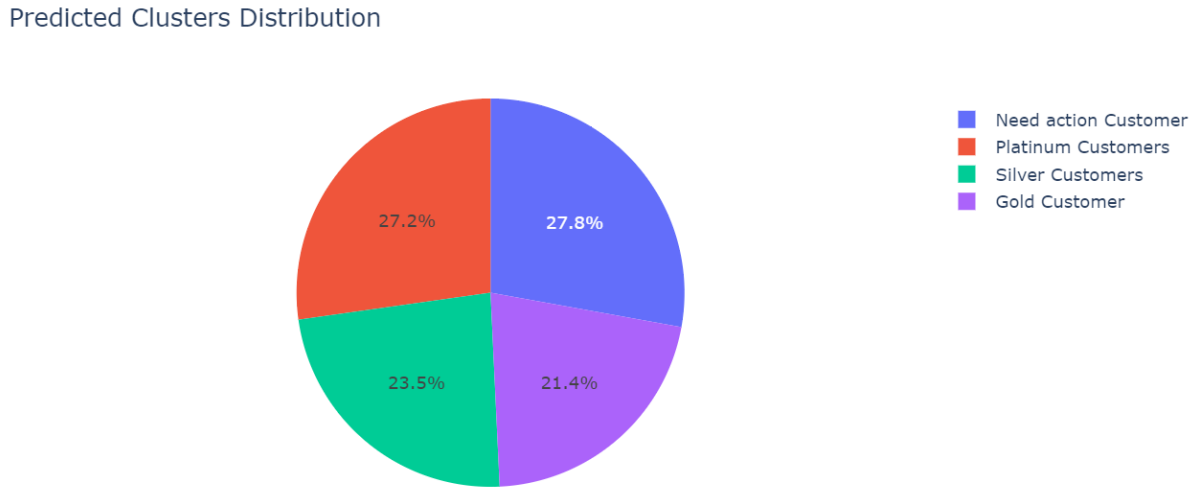
Customer ID	recency	Frequency	Monetary	Recency_Score	Frequency_Score	Monetary_Score	RFM_Score	Segment	Cluster
12346	325.0	3	77352.96	2	2	4	224	Top Customer Needed Attention	3
12347	2.0	8	5633.32	4	3	4	434	Top Recent Customer	0
12348	75.0	5	1658.40	3	3	3	333	Top Loyal Customer	3
12349	18.0	3	3678.69	4	2	4	424	Top Recent Customer	0
12350	310.0	1	294.40	2	1	1	211	Customer Needed Attention	1

Based on the clusters formed and the segments from RFM score we try to label the clusters.

Cluster	Segment	
0	Star Customer	842
	Top Recent Customer	402
	Top Loyal Customer	325
	Top Customer Needed Attention	16
	Recent Customer	8
1	Lost Customer	1032
	Customer Needed Attention	502
	Loyal Customer	43
	Top Lost Customer	30
	Top Customer Needed Attention	22
2	Loyal Customer	491
	Recent Customer	371
	Top Loyal Customer	174
	Top Recent Customer	159
	Customer Needed Attention	52
	Top Customer Needed Attention	6
3	Top Customer Needed Attention	506
	Customer Needed Attention	276
	Top Lost Customer	250
	Top Loyal Customer	165
	Lost Customer	133
	Star Customer	29
	Loyal Customer	18

Based on the above data we label the clusters as **Platinum customer, Gold customer, Silver customer, and customer needed action.**

Customer ID	recency	Frequency	Monetary	Recency_Score	Frequency_Score	Monetary_Score	RFM_Score	Segment	Cluster	Labels
12346	325.0	3	77352.96	2	2	4	224	Top Customer Needed Attention	3	Silver Customers
12347	2.0	8	5633.32	4	3	4	434	Top Recent Customer	0	Platinum Customers
12348	75.0	5	1658.40	3	3	3	333	Top Loyal Customer	3	Silver Customers
12349	18.0	3	3678.69	4	2	4	424	Top Recent Customer	0	Platinum Customers



Conclusion:

- **Three different types of analysis were performed (EDA, RFM & K-means clustering).**
- **All three analysis methods offer some specific advantages and also have limitations.**
- **First RMF analysis is used to segment the customers based on their behaviour, these segments are used to label the clusters formed.**
- **From the clusters formed, we can relate it with the various demographics like geographic location, time of purchase, type of purchase and more to drive more insights.**

Note: we've tried DB-scan for clustering, but we got better results with K-Means clustering