CAPSTONE PROJECT

ONLINE RETAIL CUSTOMER SEGMENTATION

PROBLEM STATEMENT

In this project, our task is to identify major customer segments on a transnational dataset 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.

DATA DESCRIPTION

❑ **Attribute Information:**

- 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. Numeric, the day and time when each transaction was generated.
- 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.

INTRODUCTION

- Customer segmentation is the process of separating customers into groups based on their shared behavior or other attributes. The groups should be homogeneous within themselves and should also be heterogeneous to each other.

- The main goal is to identify customers that are most profitable and the ones who churned out to prevent further loss of customer by redefining company policies.

- Having large number of customers, each with different needs it is crucial to find which customer are most important for business and target them with appropriate strategy.

IMPORTING AND INSPECTING DATASET

- Used following libraries: NumPy, pandas, seaborn, matplotlib, sklearn and SciPy.
- The dataset contain 541909 records and 8 columns.
- Number of unique values in each column is as follows:

```
InvoiceNo : 25900
StockCode : 4070
Description : 4223
Quantity : 722
InvoiceDate : 23260
UnitPrice : 1630
CustomerID : 4372
Country : 38
```

- There are some missing values in the Description and CustomerID column:
  - Description – 135080 (24.93% Missing Values)
  - CustomerID – 1454 (0.27% Missing Values)
- There are 5225 duplicate rows in the dataset.

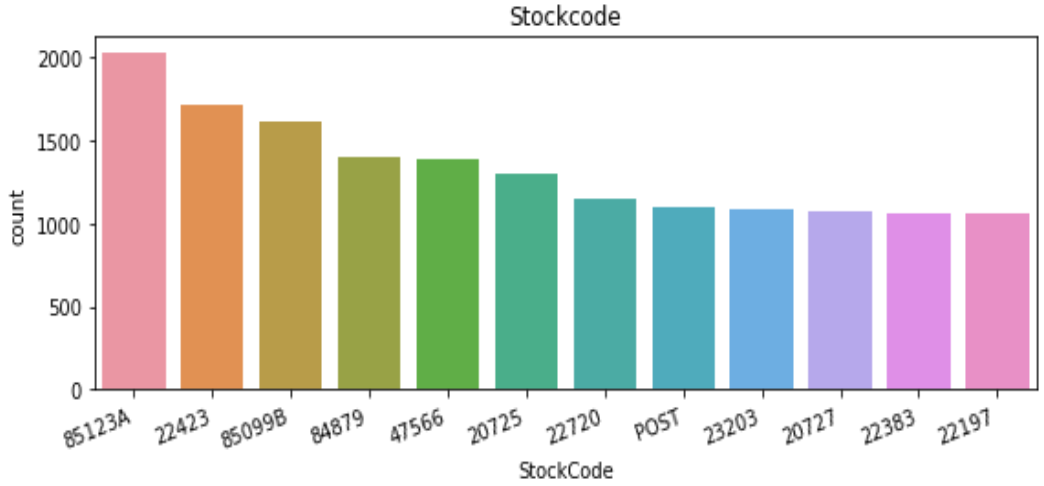
DATA CLEANING

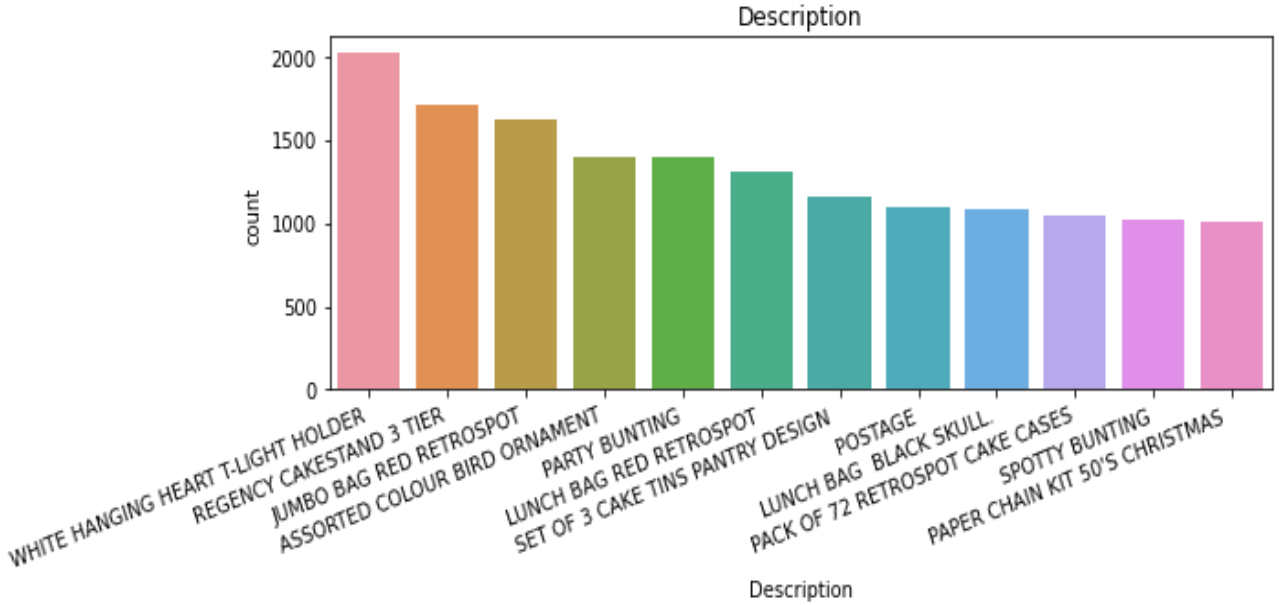
- We dropped the rows with missing values from the dataset because if the CustomerID and Description is not present, other features in that row can not be used to identify the customer hence its meaningless to assign that datapoint to any cluster.
- We also dropped the duplicate rows from the dataset so the final shape of the dataset is 401604 records with 8 columns.

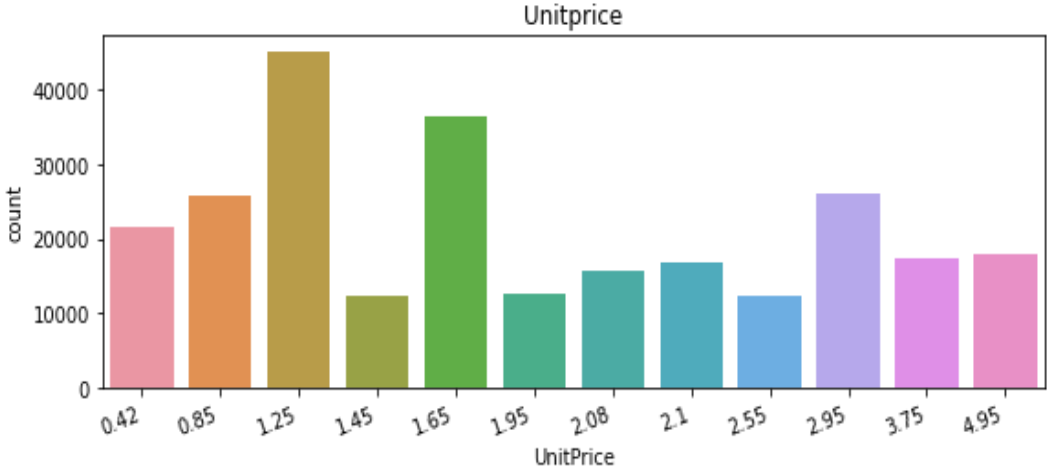
FEATURE ENGINEERING

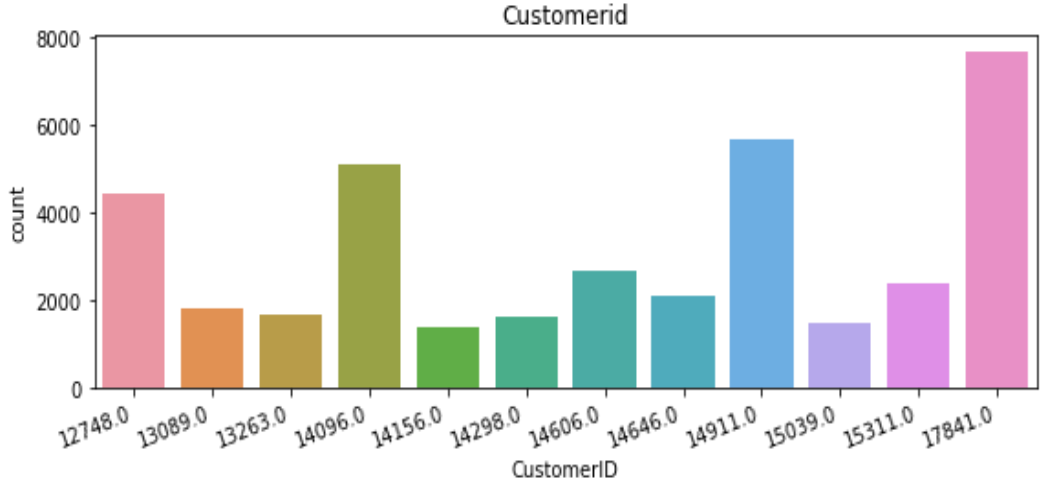
- Created some new features named Year, Month, Day and Hour by extracting different time intervals from the InvoiceDate column which was of the datetime datatype.
- Created a new feature named '**TotalAmount**' by multiplying values from **Quantity** and **UnitPrice** column.
- Created a new feature **'TimeType'** based on hours to define whether its Morning, Afternoon or Evening
- The InvoiceNo starting with 'C' represents cancellation hence we dropped all such rows as we are interested in the valid transactions only.

MOST FREQUENT VALUES

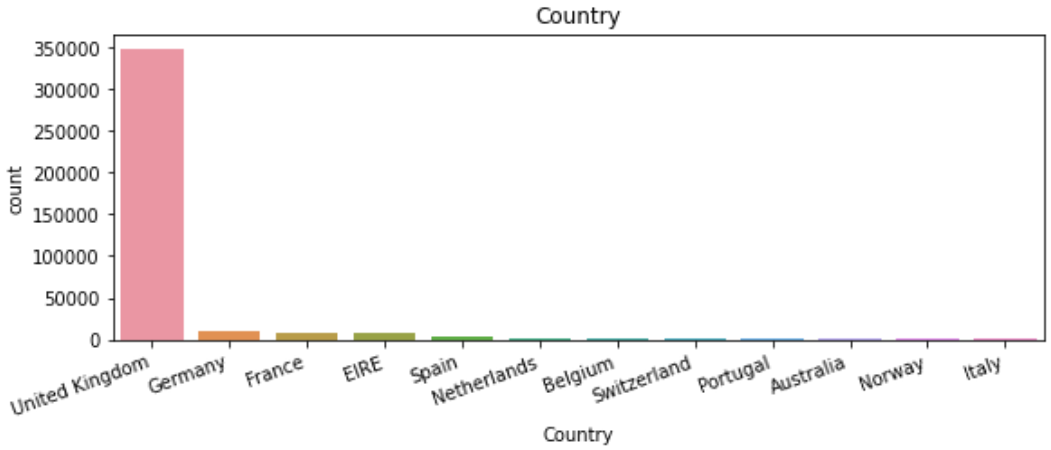


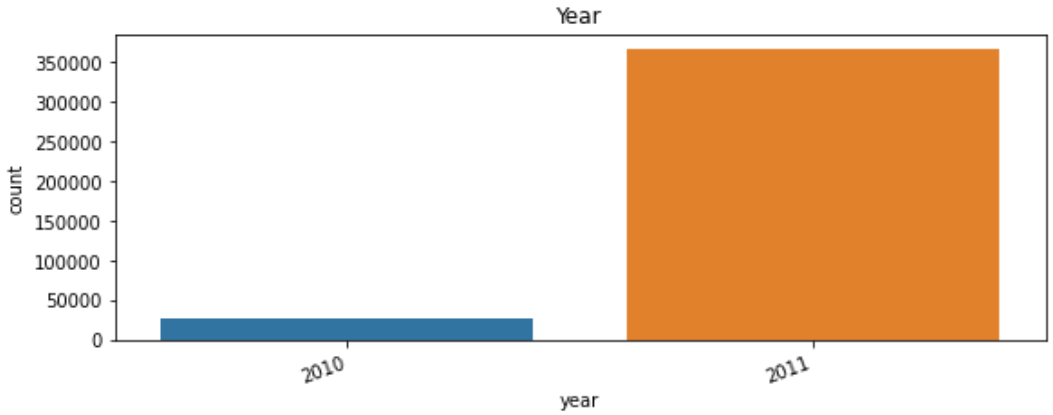


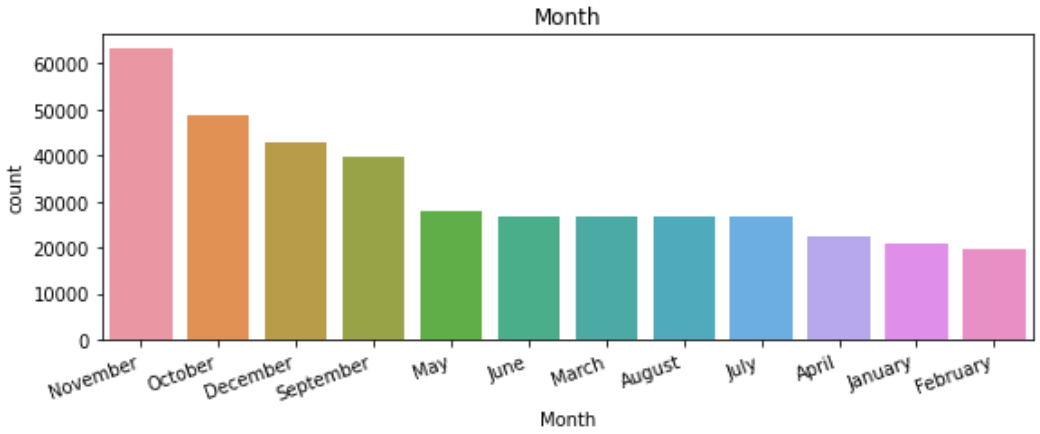


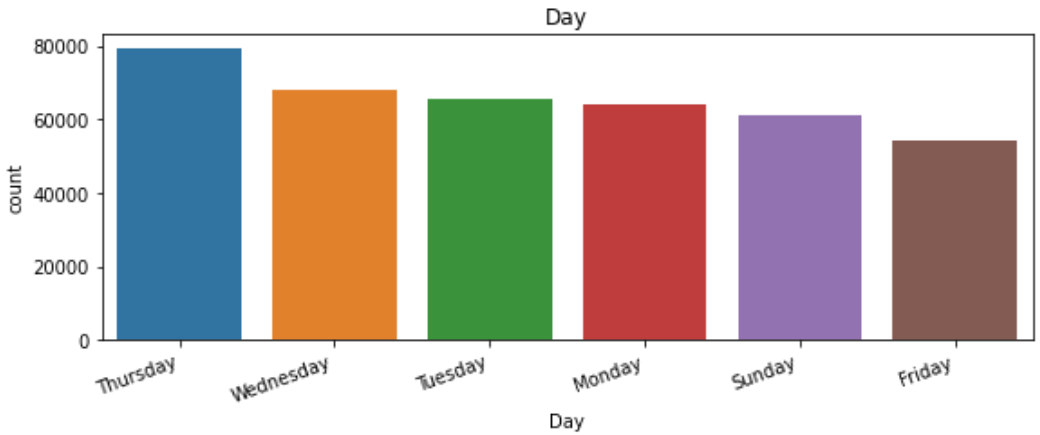


MOST FREQUENT VALUES

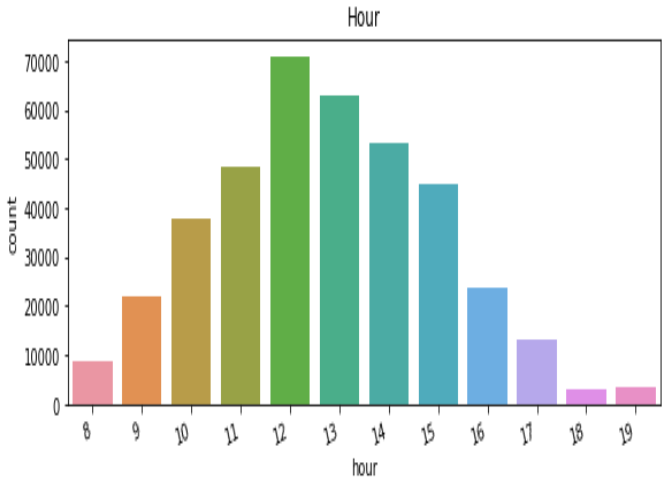


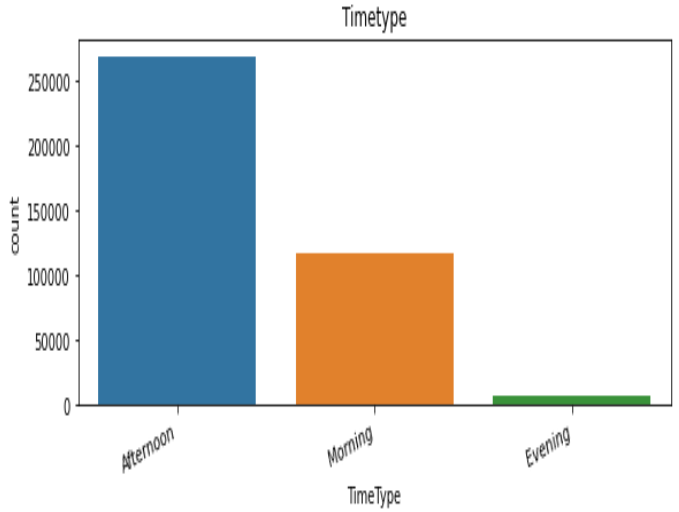


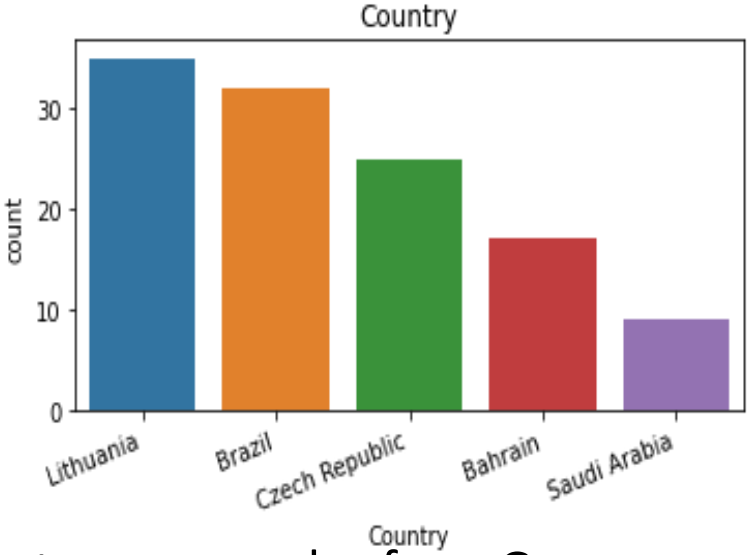




FREQUENT VALUES

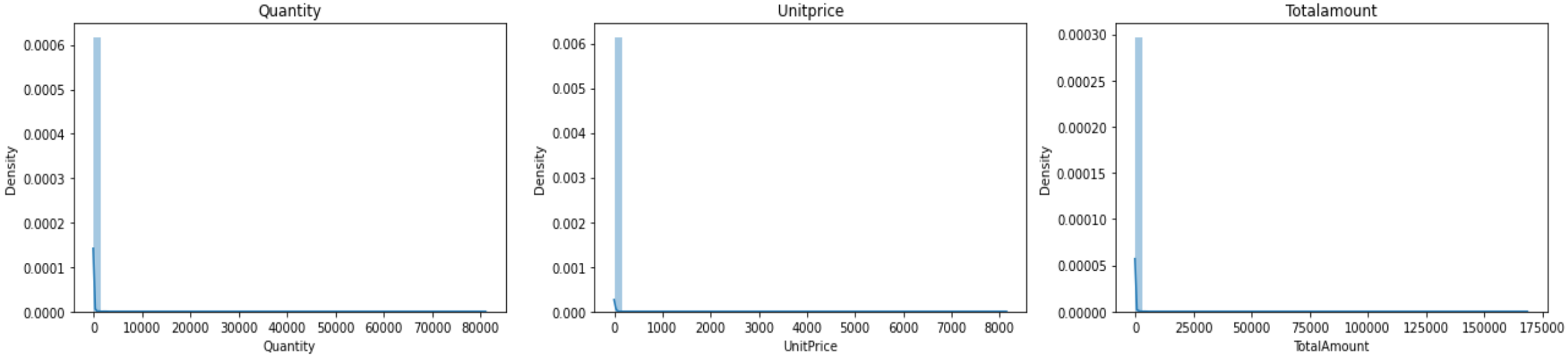






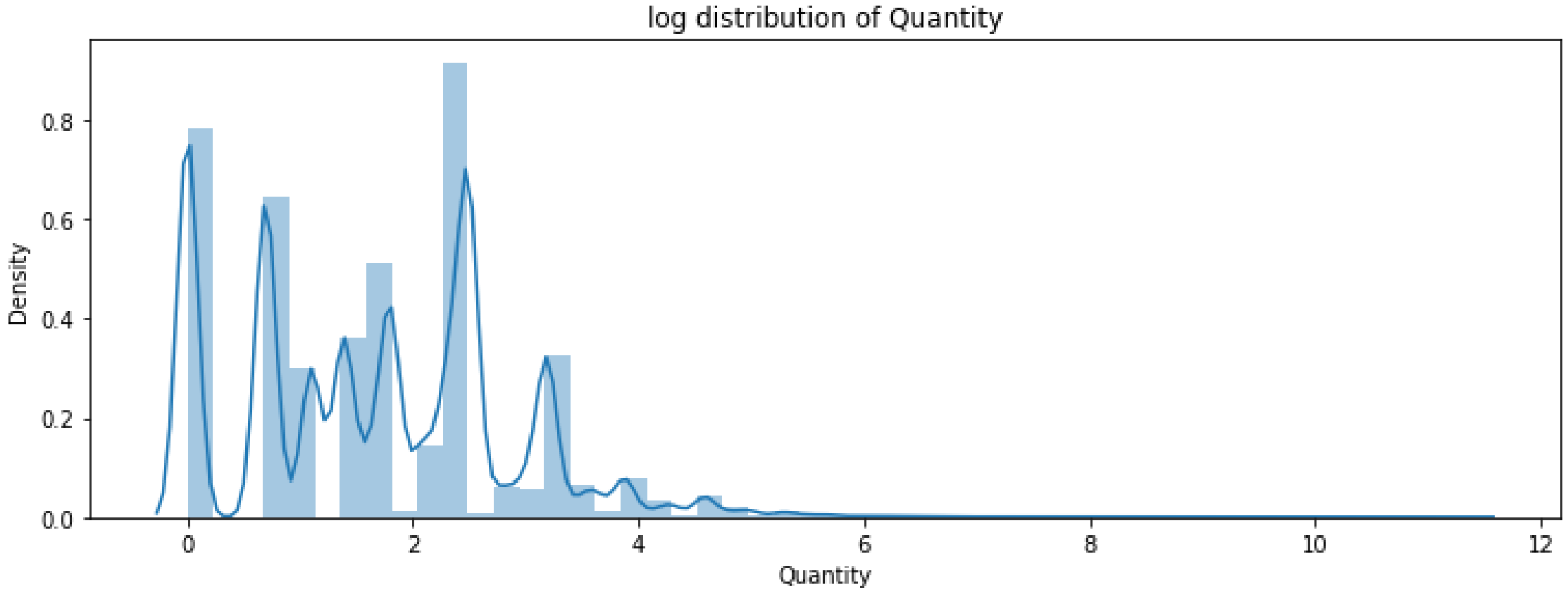
- Most Customers are from United Kingdom. Considerable number of customers are also from Germany, France, EIRE and Spain. Whereas Saudi Arabia, Bahrain, Czech Republic, Brazil and Lithuania has least number of customers.
- There are no orders placed on Saturdays. Looks like it's a non working day for the retailer.
- Most of the customers have purchased the gifts in the month of November, October, December and September. Less number of customers have purchased the gifts in the month of April, January and February.
- Most of the customers have purchased the items in Afternoon, moderate numbers of customers have purchased the items in Morning and the least in Evening.
- WHITE HANGING HEART T-LIGHT HOLDER, REGENCY CAKESTAND 3 TIER, JUMBO BAG RED RETROSPOT are the most ordered products

VISUALIZING DISTRIBUTIONS



- Visualizing the distribution of quantity, unitprice and totalamount columns
- It shows a positively skewed distribution because most of the values are clustered around the left side of the distribution while the right tail of the distribution is longer, which means mean>median>mode
- For symmetric graph mean=median=mode.

LOG TRANSFORMATION



- After applying log transformation now the distribution plot looks comparatively better than being skewed.
- We use log transformation when our original continuous data does not follow the bell curve, we can log transform this data to make it as “normal” as possible so that the analysis results from this data become more valid.

RFM MODELLING

- The idea is to segment customers based on when their last purchase was (Recency), how often they've purchased in the past (Frequency), and how much they spent (Monetary). All three of these measures have proven to be effective predictors of a customer's willingness to engage in marketing messages and offers.
- For that we created a new dataframe to calculate Recency, Frequency and Monetary scores for each customer. Here are the top five observation we got:

	CustomerID	Recency	Frequency	Monetary
0	12346.0	326	1	77183.60
1	12347.0	2	182	4310.00
2	12348.0	75	31	1797.24
3	12349.0	19	73	1757.55
4	12350.0	310	17	334.40

- We divided Recency, Frequency and Monetary columns in 4 categories based on quantiles to come up with new 3 features named R, F, and M.

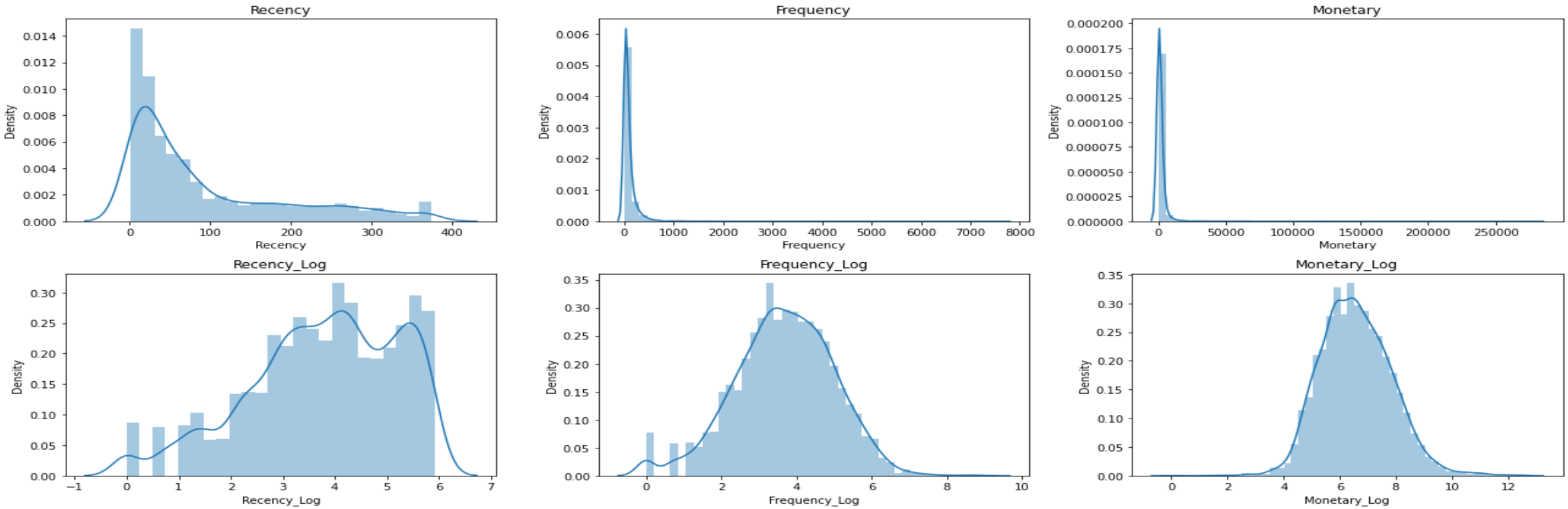
RFM MODELLING

- Finally we created a new feature named RFM by concatenating the factors R, F and M. The values in this column range from 111(lowest) to 444(highest).

CustomerID	Recency	Frequency	Monetary	R	F	M	RFM
12346.0	326	1	77183.60	1	1	4	114
12347.0	2	182	4310.00	4	4	4	444
12348.0	75	31	1797.24	2	2	4	224
12349.0	19	73	1757.55	3	3	4	334
12350.0	310	17	334.40	1	1	2	112

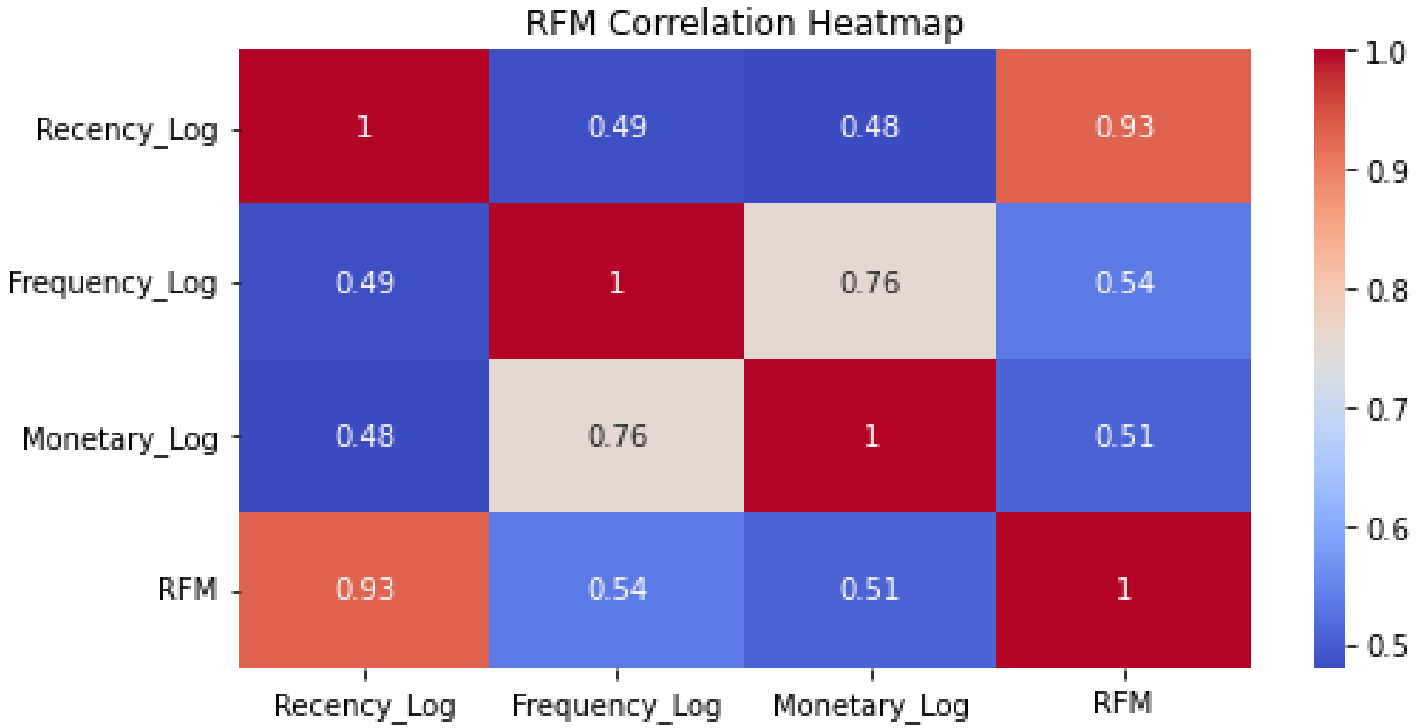
- If the RFM of any customer is 444. His Recency is good, frequency is more and Monetary is more. So, he is a Big spender.
- If the RFM of any customer is 111. His Recency is low, frequency is low and Monetary is low. So, he is a little spender.
- If the RFM of any customer is 144. He purchased a long time ago but buys frequently and spends more. And so on.

RFM MODELLING



- Earlier the distributions of Recency, Frequency and Monetary columns were positively skewed but after applying log transformation, the distributions appear to be symmetrical and normally distributed.
- It will be more suitable to use the transformed features for better visualization of clusters.

RFM CORRELATION HEATMAP

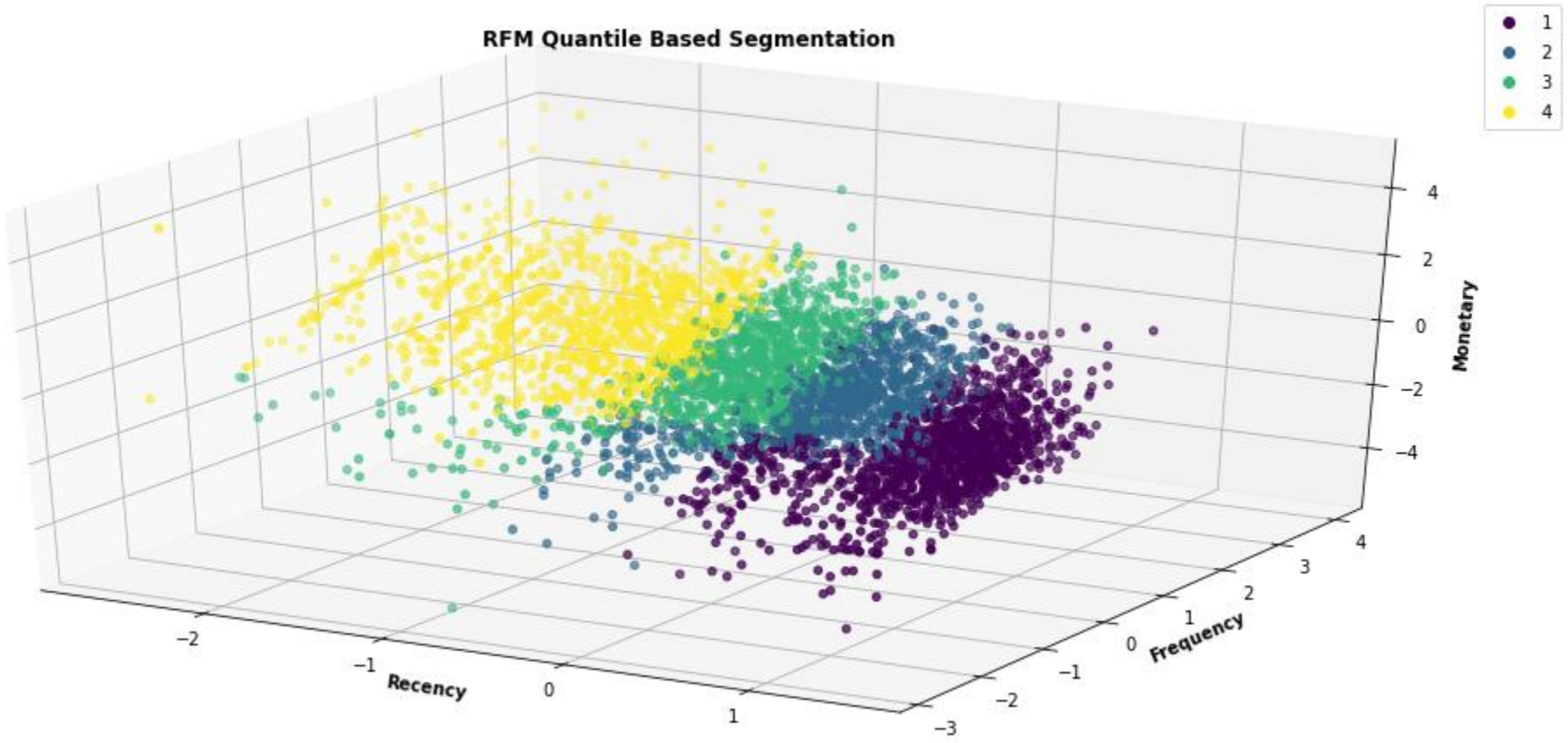


- We can see that Recency is highly correlated with the RFM value.
- Frequency and Monetary are moderately correlated with the RFM.

CLUSTERING PREREQUISITES

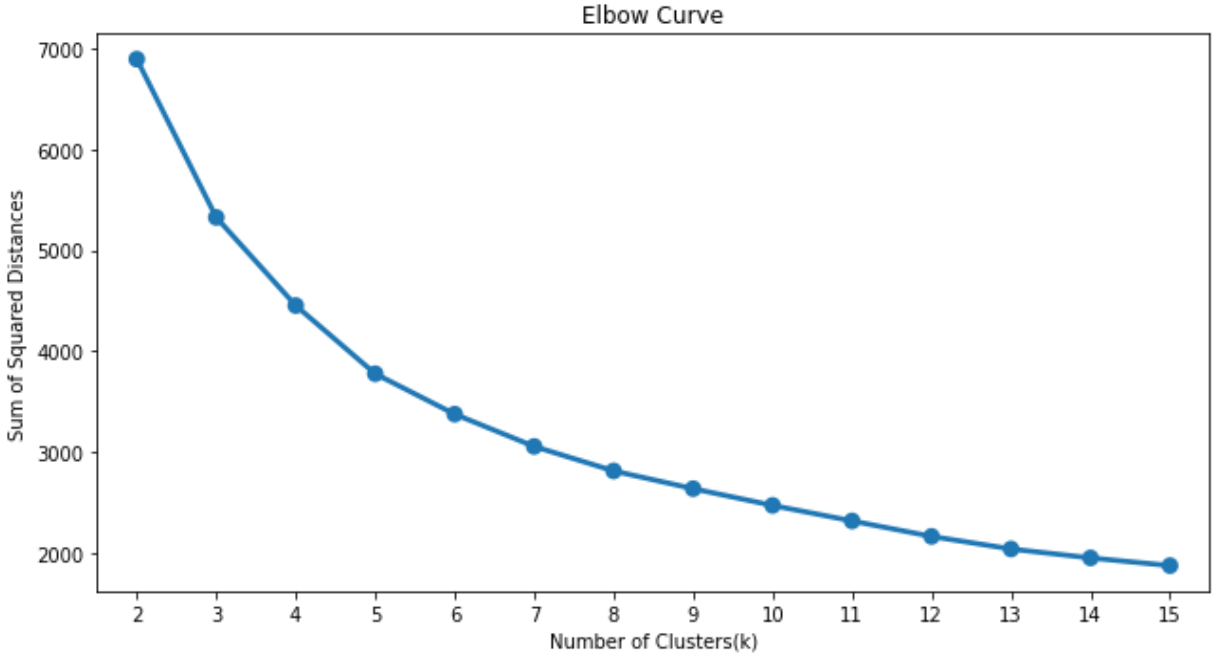
- Defining the X variables i.e. the log transformed Recency, Frequency and Monetary values.
- Applied StandardScaler on X variables. Standard Scaler helps to get standardized distribution, with a zero mean and standard deviation of one (unit variance). It standardizes features by subtracting the mean value from the feature and then dividing the result by feature standard deviation.
- Defining a function which takes predicted y labels as input and plots 3d visualization of clusters.

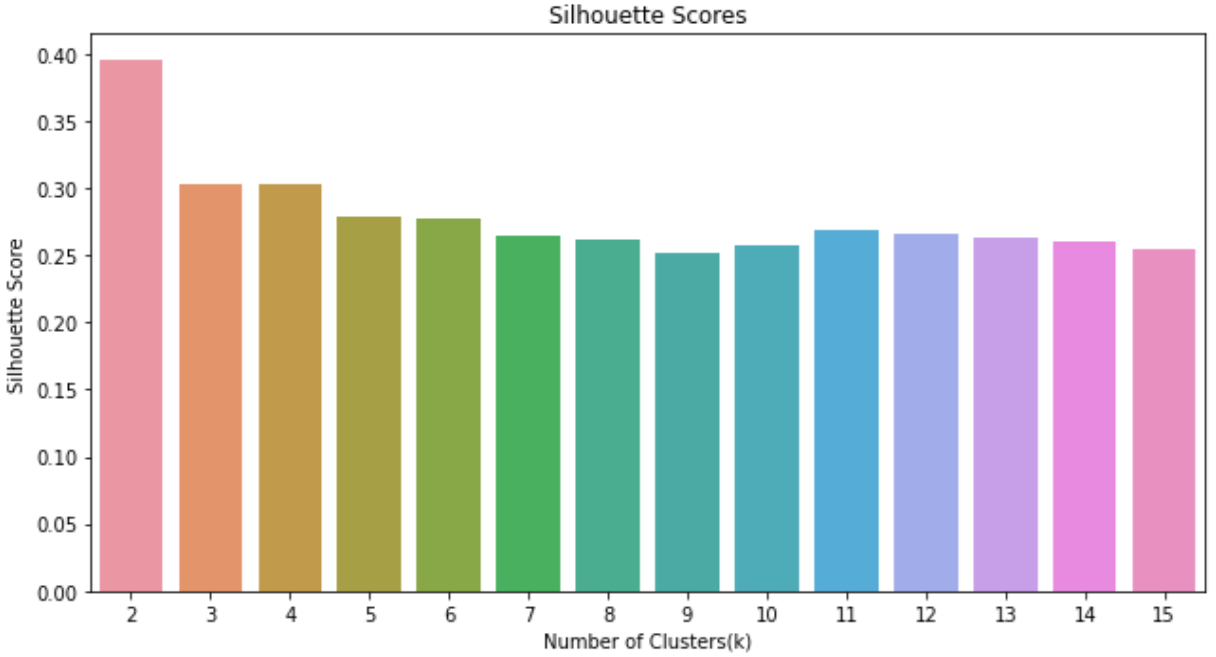
QUANTILE BASED CLUSTERING



- The values in the column RFM range from 111(lowest) to 444(highest). We divided the RFM values into 4 segments based on quantiles to categorize the customers.
- Customers who fall in the violet region have high recency whereas customers who are in the yellow region have low recency values.

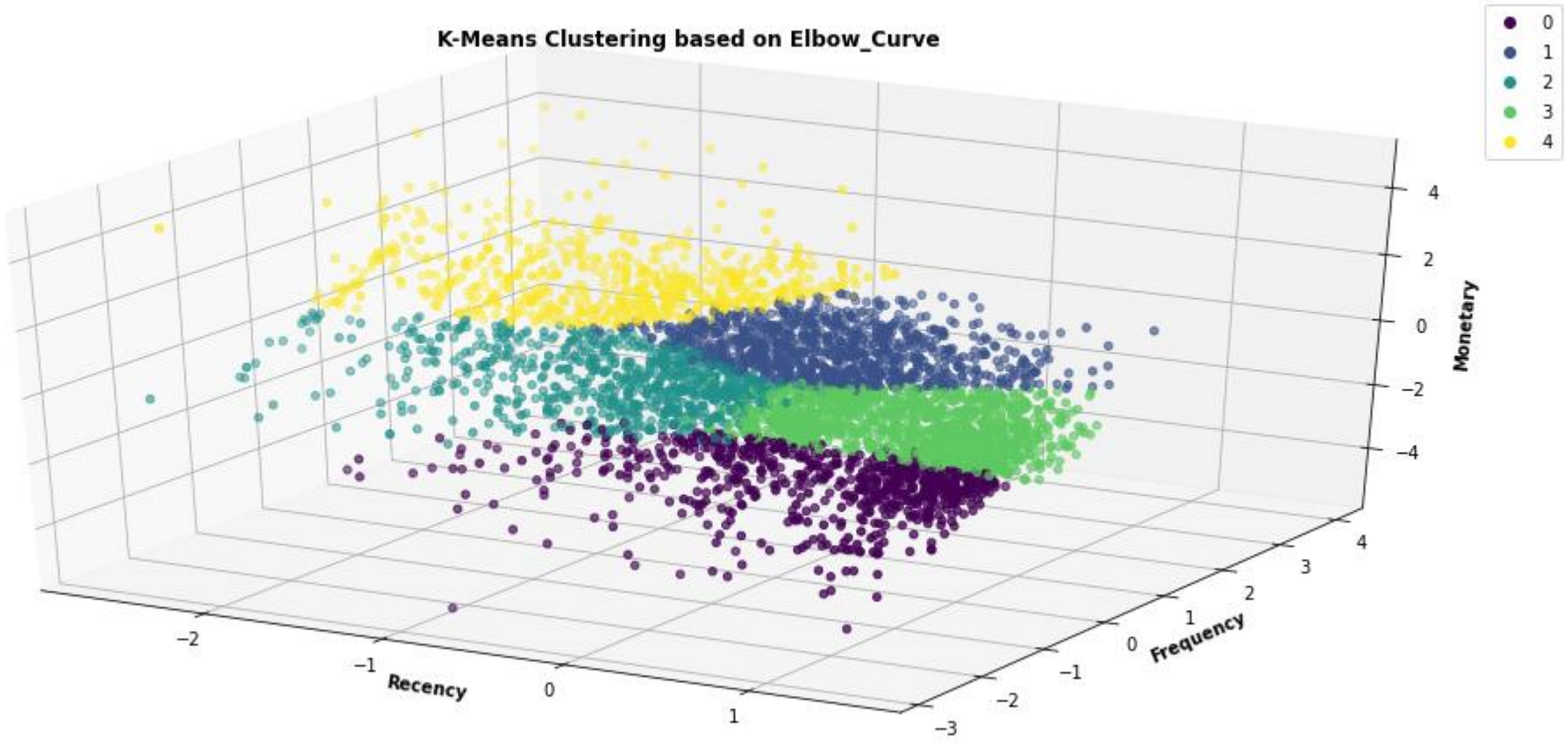
K-MEANS CLUSTERING





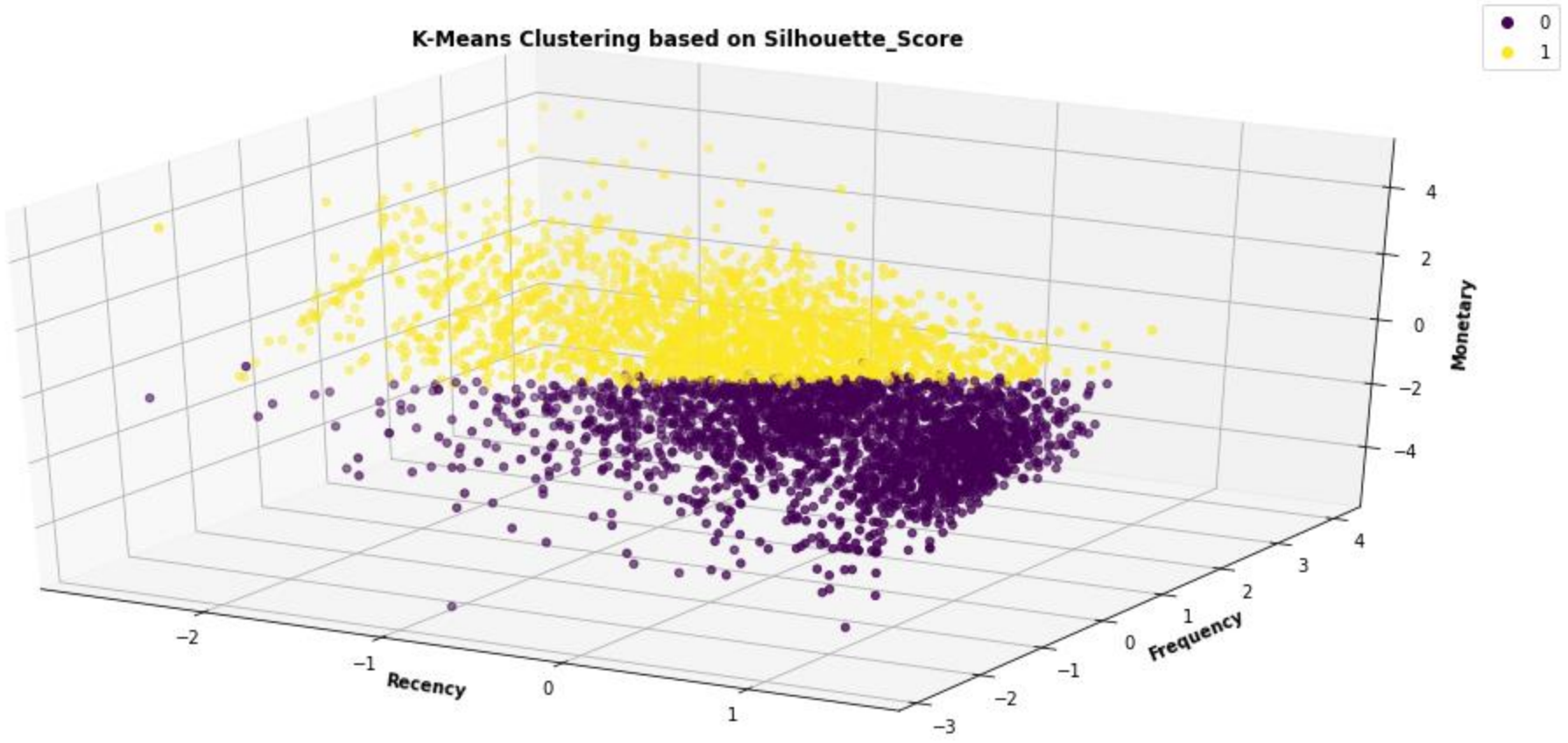
- From the Elbow curve 5 appears to be at the elbow and hence can be considered as the number of clusters. n_clusters=4 or 6 can also be considered.
- If we go by maximum Silhouette Score as the criteria for selecting optimal number of clusters, then n_clusters=2 can be chosen. 3 and 4 is also a good choice if we want more segments.
- If we look at both of the graphs at the same time to decide the optimal number of clusters, So 4 appears to be a good choice, having a decent Silhouette score as well as near the elbow of the elbow curve.

K-MEANS CLUSTERING



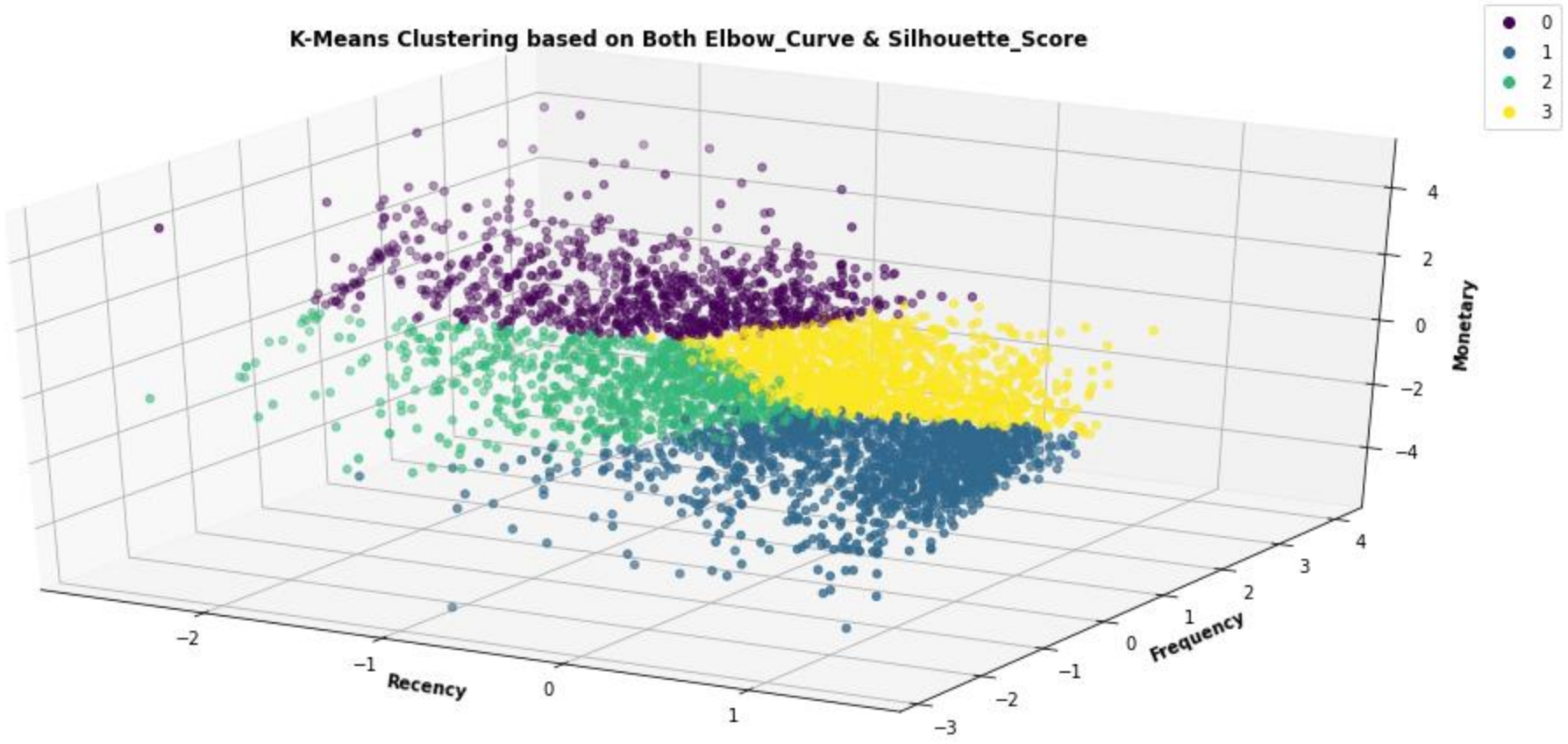
(n_clusters=5)

K-MEANS CLUSTERING



(n_clusters=2)

K-MEANS CLUSTERING



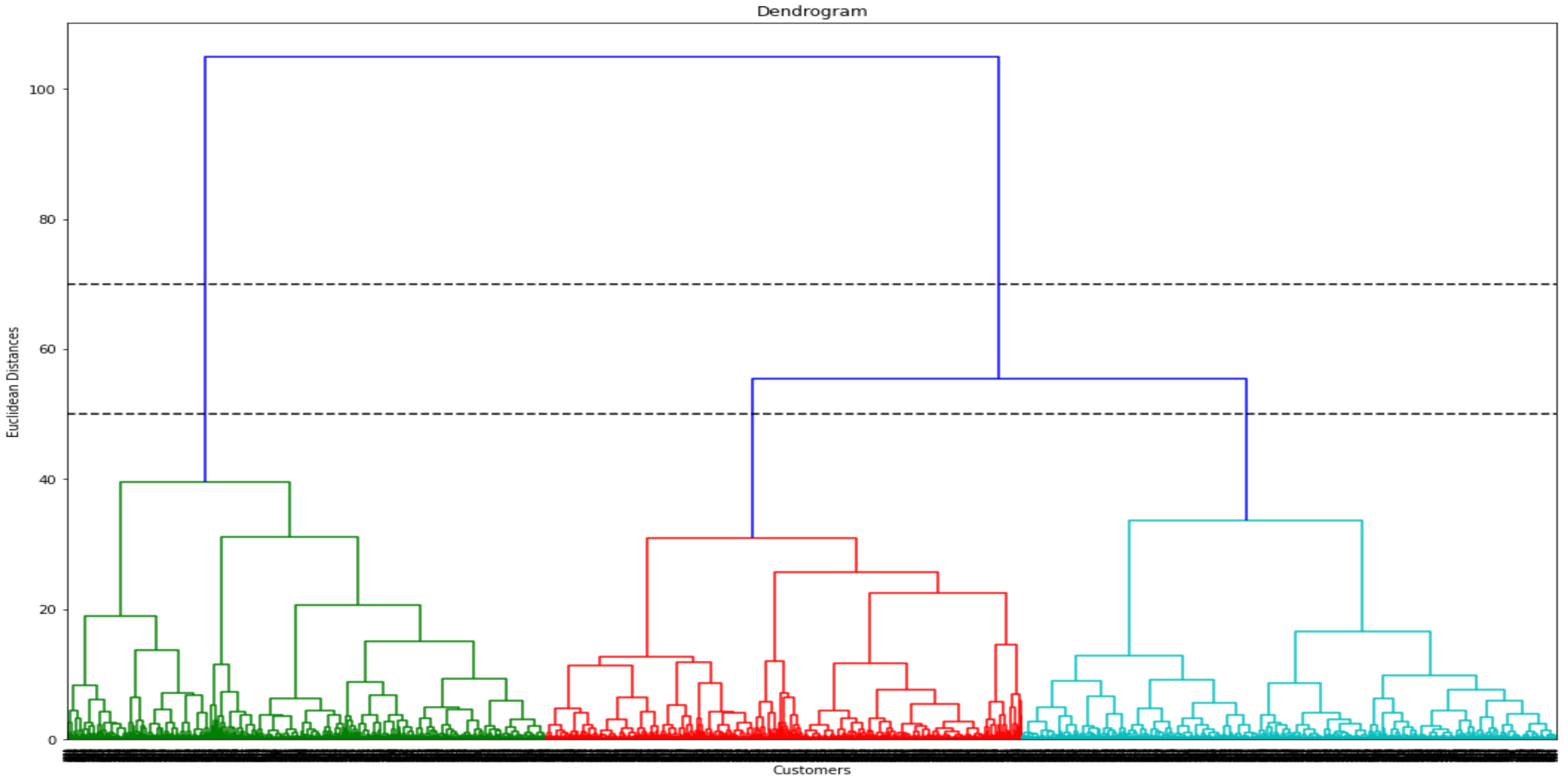
(n_clusters=4)

HIERARCHICAL CLUSTERING

AI

- In the K-means clustering there is a challenge to predetermine the number of clusters, and it always tries to create the clusters of the same size. To solve these two challenges, we can opt for the hierarchical clustering algorithm because, in this algorithm, we don't need to have knowledge about the predefined number of clusters. Hierarchical clustering is based on two techniques:
  - Agglomerative: Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
  - Divisive**:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**
- We define the optimal number of clusters based on dendogram as shown in the next slide.

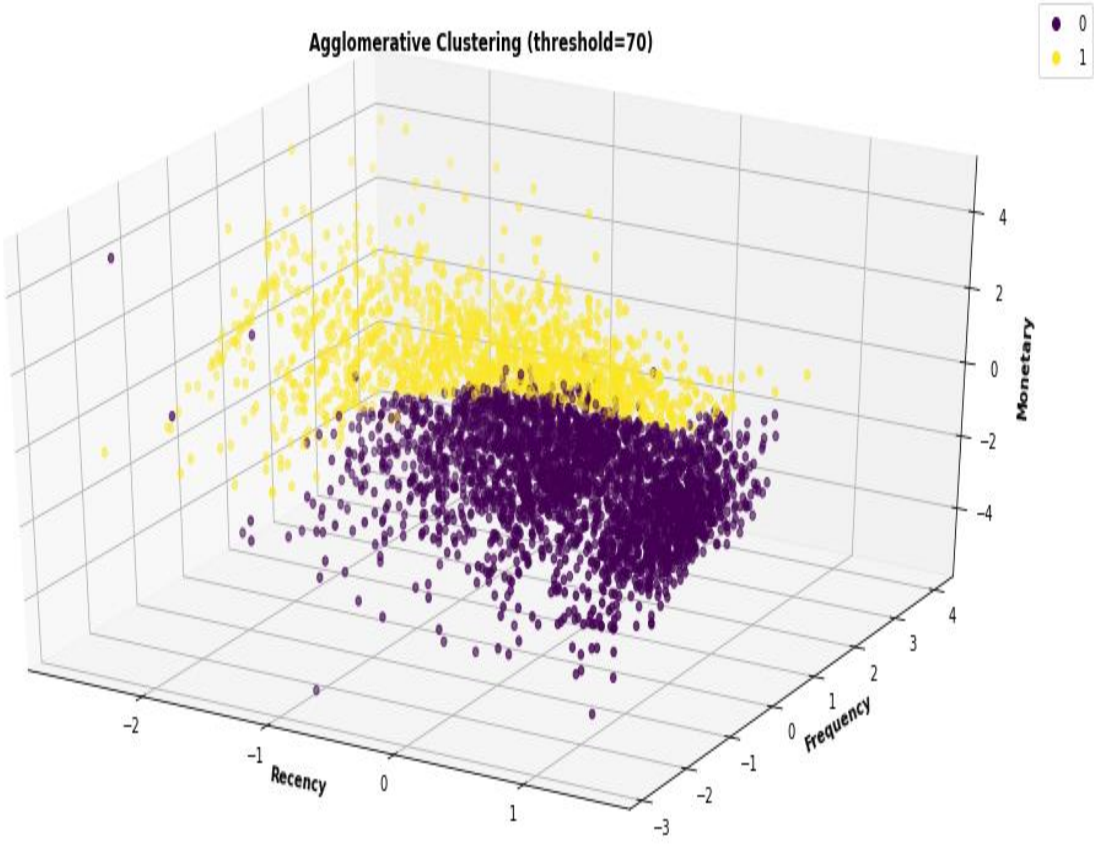
DENDOGRAM



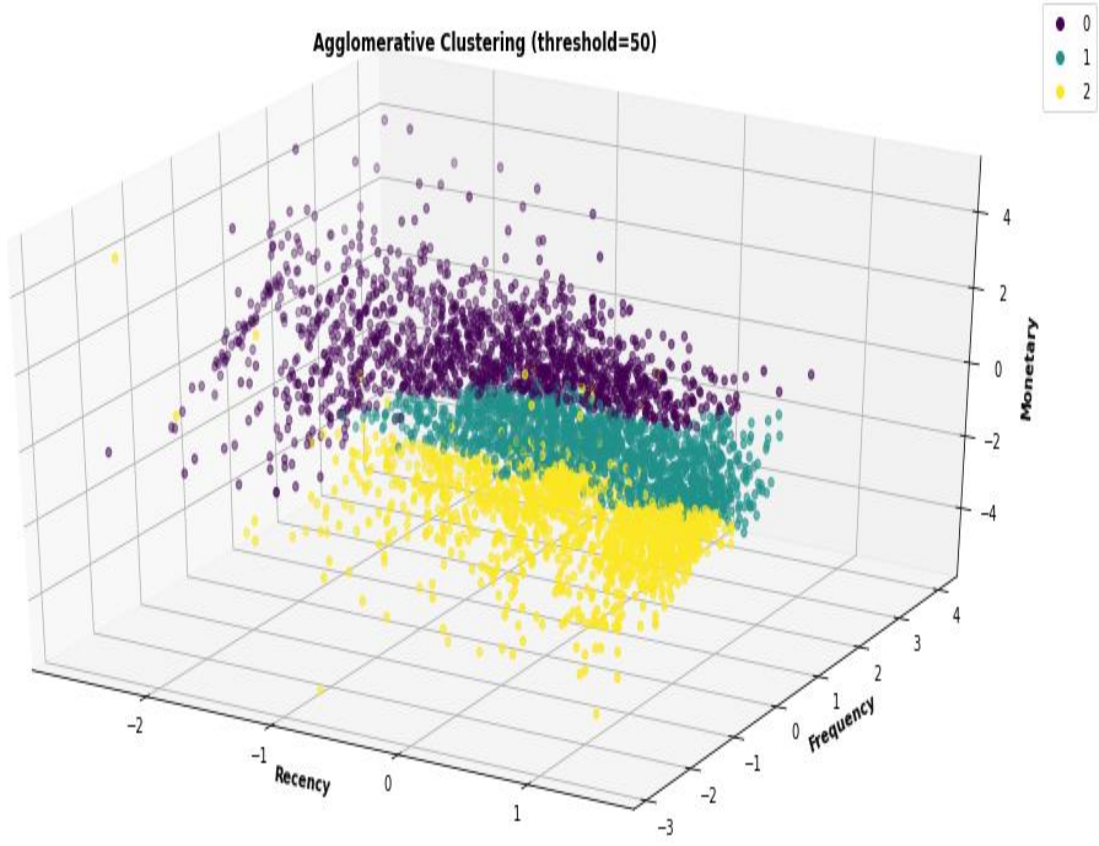
HIERARCHICAL CLUSTERING

- We can set a threshold distance and draw a horizontal line (Generally, we try to set the threshold in such a way that it cuts the tallest vertical line). We can set this threshold as 50 or 70 and draw a horizontal line as shown in dendogram in the previous slide..
- The number of clusters will be the number of vertical lines which are being intersected by the line drawn using the threshold. The larger threshold (y=70) results in 2 clusters while the smaller (y=50) results in 3 clusters.
- Lets visualize the clusters in the next slide.

HIERARCHICAL CLUSTERING

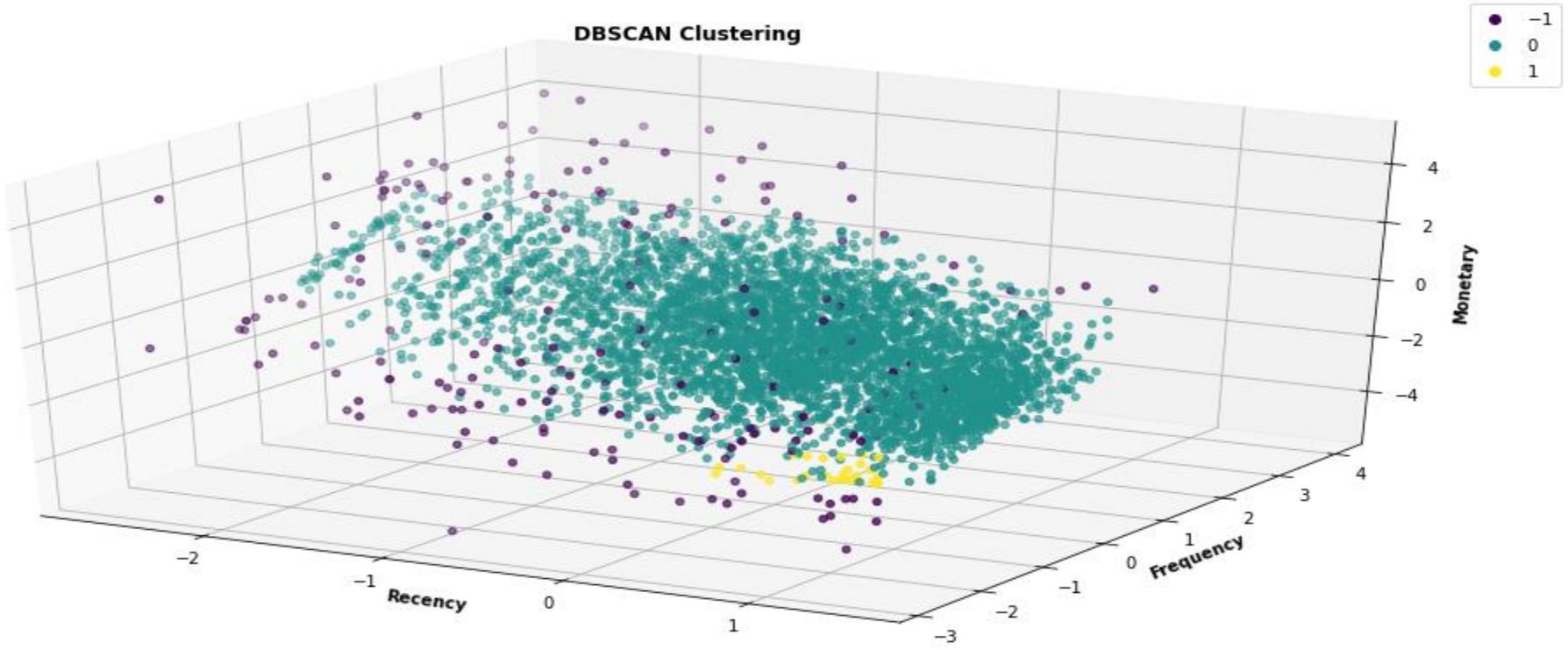


(y=70)



(y=50)

DBSCAN CLUSTERING



- Density-based spatial clustering of applications with noise (DBSCAN) is an alternative to KMeans and hierarchical clustering. It does not require us to specify the number of clusters, as the clusters are formed by a process of linking neighbor points together.
- It avoids outliers and identifies nested clusters within the data. The data is muddled and does not have major visible nested cluster, yet it has identified 3 clusters as shown above based on the hyperparameters defined.

CONCLUSION

- We started with a quantile based simple segmentation model first then moved to more complex models because simple implementation helps having a first glance at the data and know where/how to exploit it better.
- Then we moved to k-means clustering and visualized the results with different number of clusters. As we know there is no assurance that k-means will lead to the global best solution. We moved forward and tried Hierarchical Clustering and DBSCAN clusterer as well.
- We didn't obtain a clearly separated clusters as the Cluster assignments are muddled. Segments depend on how the business plans to use the results, and the level of granularity they want to see in the clusters. Based on that various methods of clustering can be further exploited whether applied on RFM variables or directly on the transactional dataset.

CONCLUSION

- Keeping these points in view we clustered the major segments based on our understanding as per different criteria as shown in the table below.

	Clusterer	Criterion	Segments
0	Quantile Cut	Quantile	4
1	K-Means	Elbow Curve	5
2	K-Means	Silhouette Score	2
3	K-Means	Elbow Curve & Silhouette Score	4
4	Agglomerative	Dendogram (y=70)	2
5	Agglomerative	Dendogram (y=50)	3
6	DBSCAN	NaN	3