UNSUPERVISED MACHINE LEARNING CAPSTONE PROJECT

ONLINE RETAIL CUSTOMER SEGMENTATION

Presented by-

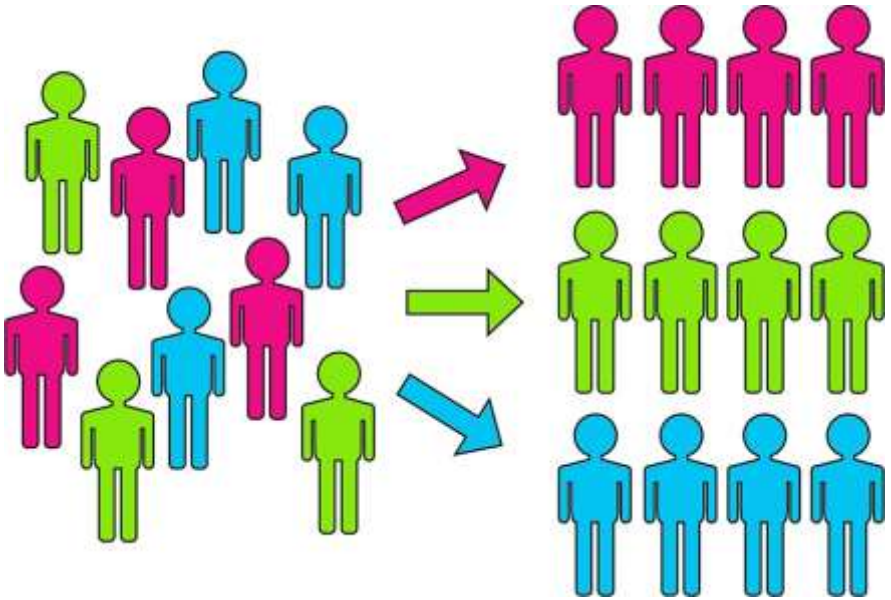
MANISH GUPTA

Contents:

- Introduction
- Problem Statement
- Data Analysis Steps
- Data Preview
- Data Summary
- RFM Modeling
- MinMaxScalar
- Clustering
- Conclusion

Introduction:

Customer segmentation is the process of separating customers into groups on the basis of their shared behavior or other attributes. The groups should be homogeneous within themselves and should also be heterogeneous to each other. The overall aim of this process is to identify high-value customer base i.e. customers that have the highest growth potential or are the most profitable. Insights from customer segmentation are used to develop tailor-made marketing campaigns and for designing overall marketing strategy and planning. A key consideration for a company would be whether or not to segment its customers and how to do the process of segmentation. This would depend upon the company philosophy and the type of product or services it offers. The type of segmentation criterion followed would create a big difference in the way the business operates and formulates its strategy.



Problem Statement:

- Customer segmentation has a lot of potential benefits. It helps a company to develop an effective strategy for targeting its customers. This has a direct impact on the entire product development cycle, the budget management practices, and the plan for delivering targeted promotional content to customers. For example, a company can make a high-end product, a budget product, or a cheap alternative product, depending upon whether the product is intended for its most high yield customers, frequent purchasers or for the low-value customer segment

- Finally, this technique can also be used by companies to test the pricing of their different products, improve customer service, and upsell and cross-sell other products or services.

Data Analysis Steps:

Imported Libraries

In this part, we imported the required libraries NumPy, Pandas, matplotlib, and seaborn, to perform Exploratory Data Analysis and for prediction, we imported the Scikit learn library.

Descriptive Statistics

In this part, we start by looking at descriptive statistic parameters for the dataset. We will use describe() this told mean, median, standard deviation

Missing Value Imputation

We will now check for missing values in our dataset. after checking not existed any missing values, In case there are any missing entries, we will impute them with appropriate values.

Graphical Representation

We will start with Univariate Analysis, bivariate Analysis and conclude with various prediction models helps us predict the Risk.

Dataset Preview:

- **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.

Data Summary :

The dataset contains 8 columns and 541909 rows.

There also exist some null values in our data:

- Percentage of null values in Description : 0.26/%
- Percentage of null values in CustomerID : 24.9%

	DataType	Non-null_Values	Unique_Values	NaN_Values	NaN_Values_Percentage
InvoiceNo	object	541909	25900	0	0.000000
StockCode	object	541909	4070	0	0.000000
Description	object	540455	4223	1454	0.268311
Quantity	int64	541909	722	0	0.000000
InvoiceDate	datetime64[ns]	541909	23260	0	0.000000
UnitPrice	float64	541909	1630	0	0.000000
CustomerID	float64	406829	4372	135080	24.926694
Country	object	541909	38	0	0.000000

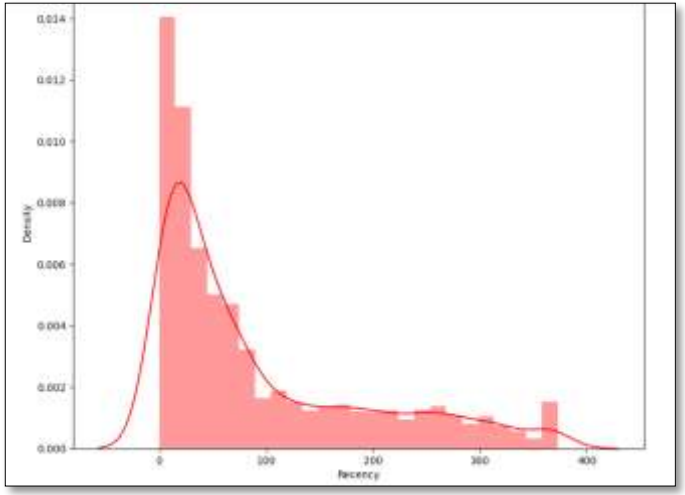
RFM Modeling

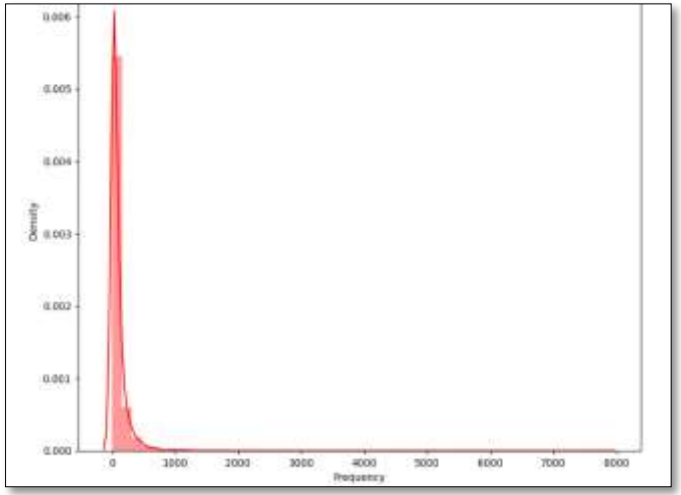
- RFM stands for Recency, Frequency, and Monetary. RFM analysis is a commonly used technique to generate and assign a score to each customer based on how recent their last transaction was (Recency), how many transactions they have made in the last year (Frequency), and what the monetary value of their transaction was (Monetary).

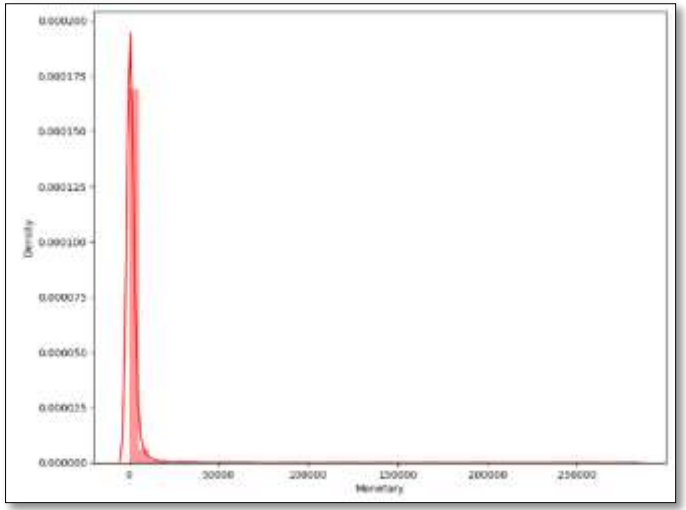
- RFM analysis helps to answer the following questions: Who was our most recent customer? How many times has he purchased items from our shop? And what is the total value of his trade? All this information can be critical to understanding how good or bad a customer is to the company.

	CustomerID	Recency	Frequency	Monetary
0	12346.0	325	1	77183.60
1	12347.0	2	182	4310.00
2	12348.0	75	31	1797.24
3	12349.0	18	73	1757.55
4	12350.0	310	17	334.40

- With the help of histogram we can say that Recency is right skewed where as Frequency and Monetary are left skewed



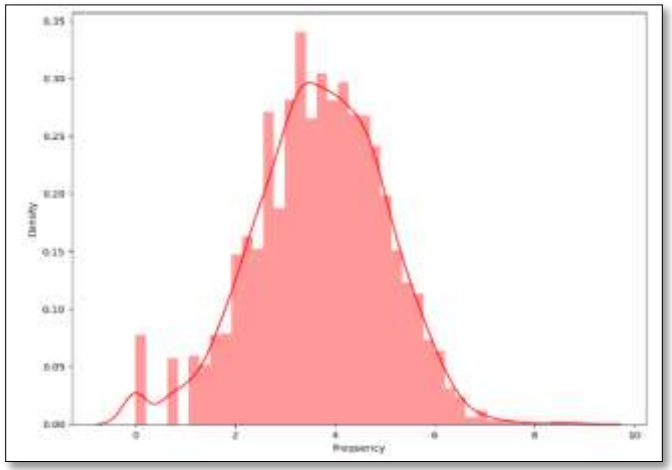


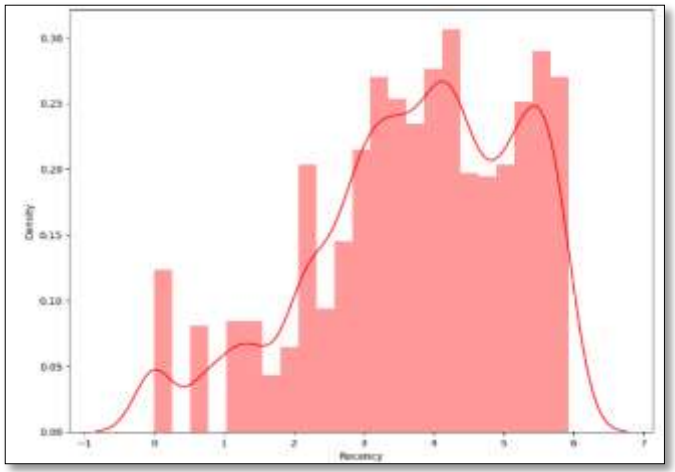


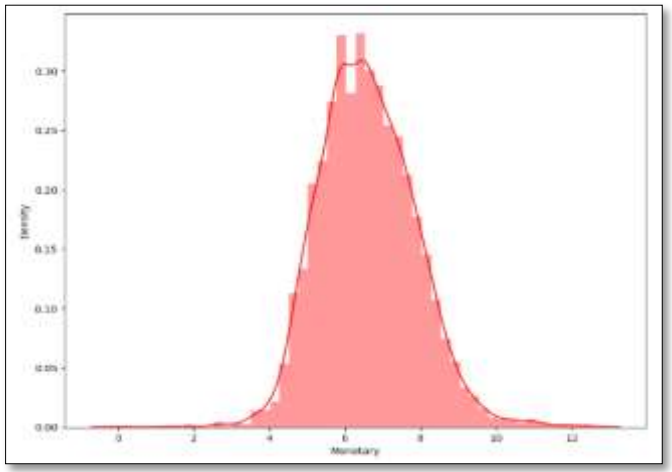
Log Transformation

#Perform Log transformation to bring data into normal or near normal distribution

```
Log_Data = RFMScores[['Recency', 'Frequency', 'Monetary']].apply(np.log, axis = 1).round(3)
```







After getting the RFM values, a common practice is to create ‘quartiles’ on each of the metrics and assigning the required order. For example, suppose that we divide each metric into 4 cuts. For the recency metric, the highest value, 4, will be assigned to the customers with the least recency value (since they are the most recent customers). For the frequency and monetary metric, the highest value, 4, will be assigned to the customers with the Top 25% frequency and monetary values, respectively. After dividing the metrics into quartiles, we can collate the metrics into a single column (like a string of characters {*like ‘213’*}) to create classes of RFM values for our customers. We can divide the RFM metrics into lesser or more cuts depending on our requirements

CustomerID	Recency	Frequency	Monetary	R	F	M	RFMGroup	RFMScore
12346.0	325	1	77183.60	4	4	1	441	9
12347.0	2	182	4310.00	1	1	1	111	3
12348.0	75	31	1797.24	3	3	1	331	7
12349.0	18	73	1757.55	2	2	1	221	5
12350.0	310	17	334.40	4	4	3	443	11

K-Mean Clustering

- KMeans requires the number of clusters to be specified during the model building process. To know the right number of clusters, methods such as silhouette analysis and elbow method can be used. These methods will help in selection of the optimum number of clusters

Applying Silhouette Score Method on Recency and Monetary

```
#silhoutte score
features_rec_mon=['Recency_log','Monetary_log']
X_features_rec_mon=rfm_df[features_rec_mon].values
scaler_rec_mon=preprocessing.StandardScaler()
X_rec_mon=scaler_rec_mon.fit_transform(X_features_rec_mon)
X=X_rec_mon
range_n_clusters = [2,3,4,5,6,7,8,9,10,11,12,13,14,15]
for n_clusters in range_n_clusters:
    clusterer = KMeans(n_clusters=n_clusters)
    preds = clusterer.fit_predict(X)
    centers = clusterer.cluster_centers_

    score = silhouette_score(X, preds)
    print("For n_clusters = {}, silhouette score is {}".format(n_cluste
```

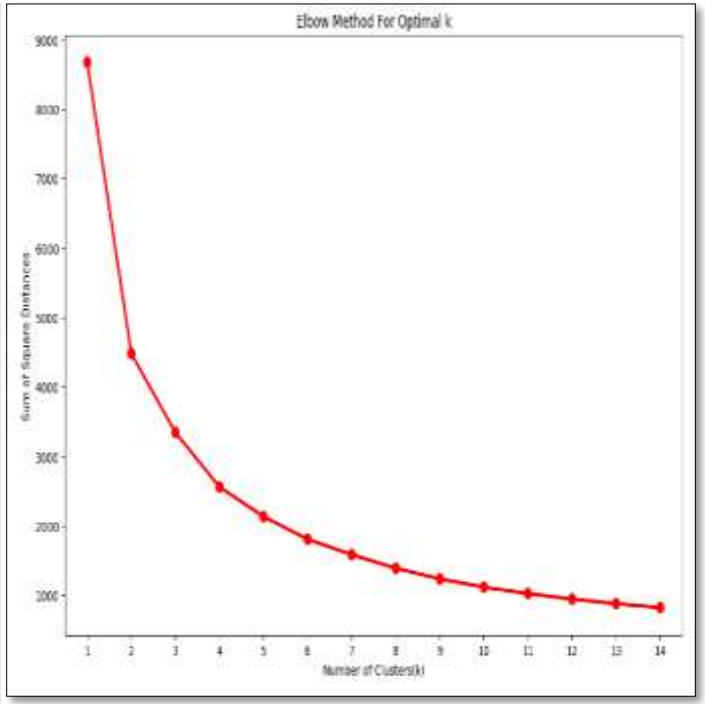
Applying Elbow Method on Recency and Monetary

```
#applying elbow method
features_rec_mon=['Recency_log','Monetary_log']
X_features_rec_mon=rfm_df[features_rec_mon].values
scaler_rec_mon=preprocessing.StandardScaler()
X_rec_mon=scaler_rec_mon.fit_transform(X_features_rec_mon)
X=X_rec_mon

from sklearn.cluster import KMeans

sum_of_sq_dist = {}
for k in range(1,15):
    km = KMeans(n_clusters= k, init= 'k-means++', max_iter= 1000)
    km = km.fit(X)
    sum_of_sq_dist[k] = km.inertia_

#Plot the graph for the sum of square distance values and Number of Clu
plt.figure(figsize=(12,8))
sns.pointplot(x = list(sum_of_sq_dist.keys()), y = list(sum_of_sq_dist.
plt.xlabel('Number of Clusters(k)')
plt.ylabel('Sum of Square Distances')
plt.title('Elbow Method For Optimal k')
plt.show()
```

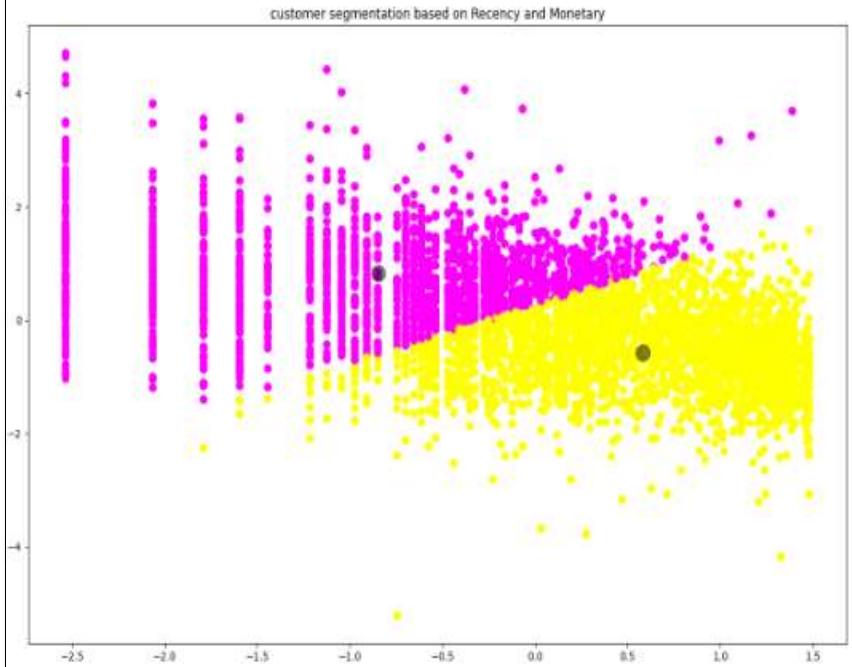


Using the Elbow Method we select the optimal number of clusters to be 3 or 4 , From the above analysis, we can see that there should be 4 clusters in our data.

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2)
kmeans.fit(X)
y_kmeans= kmeans.predict(X)
```

```
plt.figure(figsize=(15,10))
plt.title('customer segmentation based on Recency and Monetary')
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='spring_r')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
```



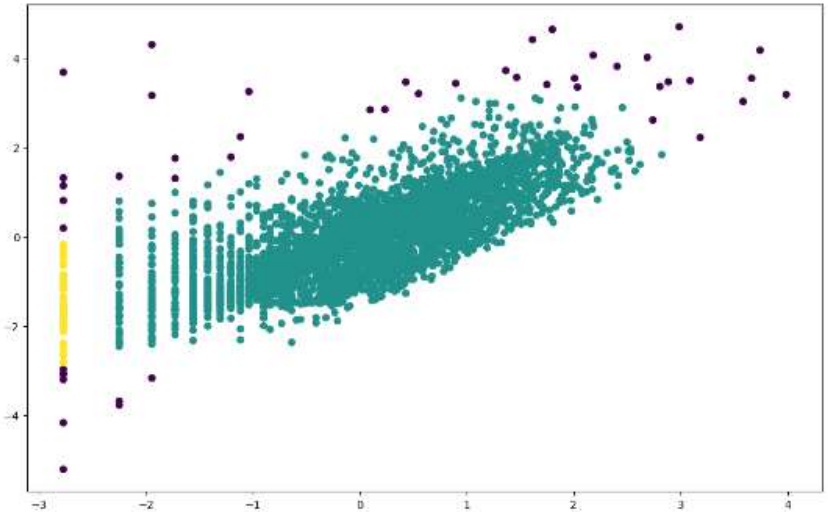
♦ *Here we can see that, Customers are well separated when we cluster them by Recency and Monetary.*

DBSCAN Clustering

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise. DBSCAN is a density-based clustering algorithm that works on the assumption that clusters are dense regions in space separated by regions of lower density.It groups ‘densely grouped’ data points into a single cluster.

Applying DBSCAN to Method on Frquency and Monetary

```
from sklearn.cluster import DBSCAN
from sklearn import metrics
y_pred = DBSCAN(eps=0.5, min_samples=15).fit_predict(X)
plt.figure(figsize=(13,8))
plt.scatter(X[:,0], X[:,1], c=y_pred);
```



Hierarchical clustering

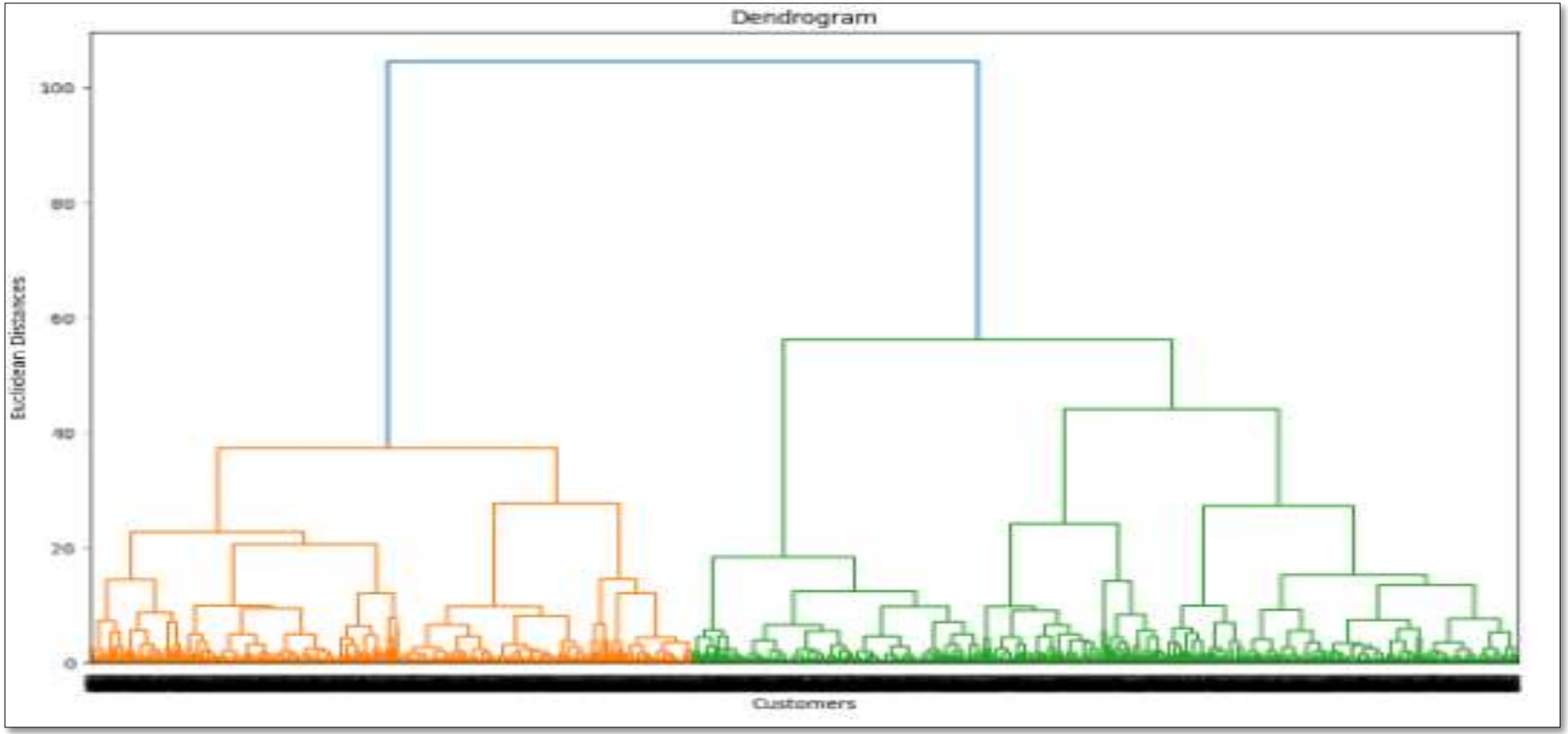
Hierarchical clustering is an unsupervised clustering algorithm which involves creating clusters that have predominant ordering from top to bottom.

Dendogram

A Dendrogram is a type of tree diagram showing hierarchical relationships between different sets of data.

```
# Using the dendogram to find the optimal number of clusters
#importing necessary library
import scipy.cluster.hierarchy as sch
# Creating a dendrogram to visualize the clusters
plt.figure(figsize=(13,8))
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean Distances')
plt.show()
```

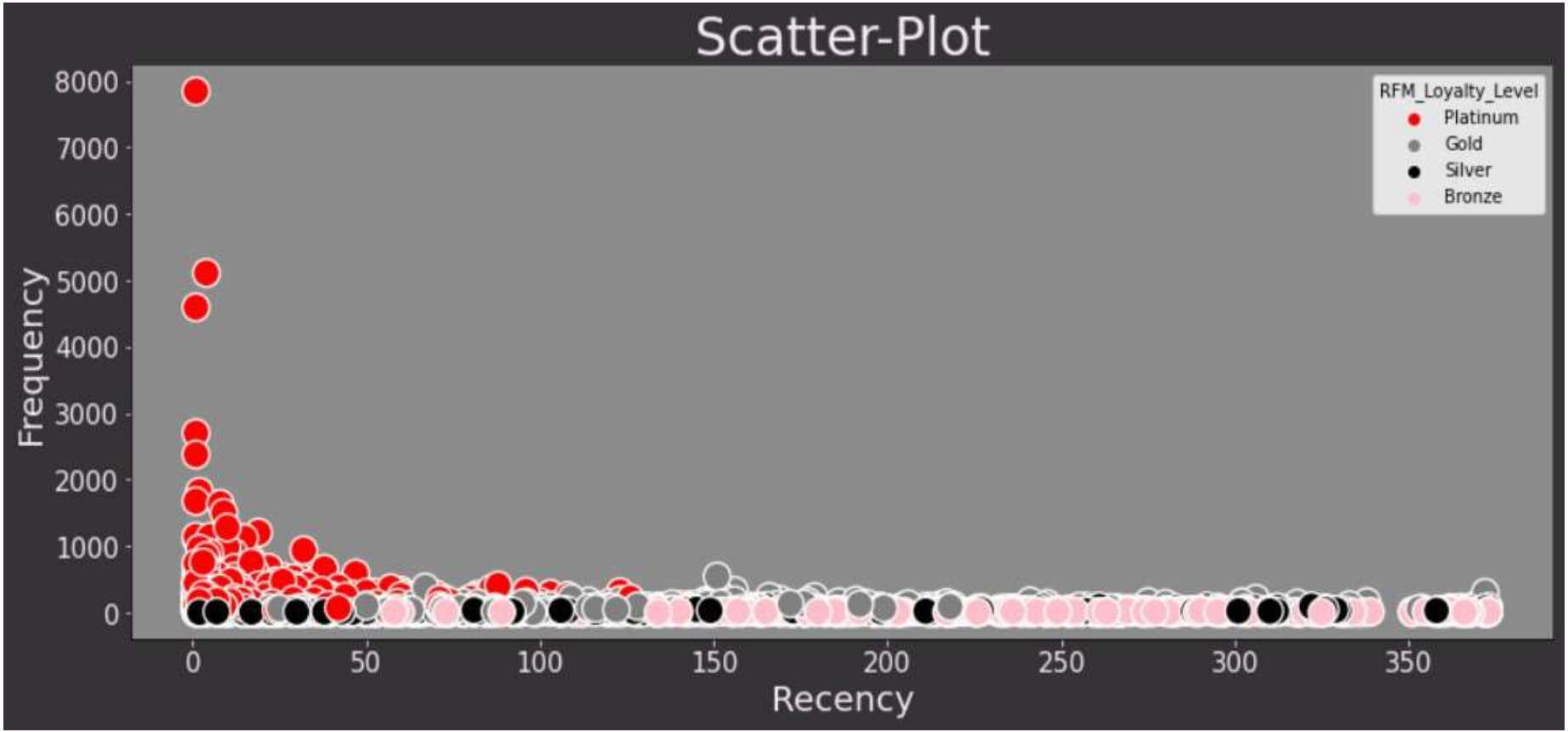
Hierarchical clustering



To understand what these 4 clusters mean in a business scenario, we should look back the table comparing the clustering performance of 3 and 4 clusters for the mean values of recency, frequency, and monetary metric. On this basis, let us label the clusters as ‘New customers’, ‘Lost customers’, ‘Best customers’, and ‘At risk customers’.

Cluster	Type of customers	RFM Interpretation	Recommended action
0	New customers	Customers who transacted recently and have lower purchase frequency, with low amount of monetary spending.	Need to handled with care by improving relationships with them. Company should try to enhance their purchasing experience by providing good quality products and services, and customer care services.
1	Lost customers	Customers with the least monetary spending and the least number of transactions. Made their last purchase long ago.	These customers may have already exited from the customer base. The company should try to understand why they left the system so that it does not happen again.
2	Best customers	Most frequent spenders with the highest monetary spending amount and had transacted recently.	Potential to be the target of new products made by a company and can increase company revenue by repeated advertising. Heavy discounts not required.
3	At risk customers	Customers who made their last transaction a while ago and made less frequent and low monetary purchases.	At high risk of churning. Need to be addressed urgently with focussed advertising. May perform well if discounts are provided to them. Company should find out why they are leaving.

Final we make 4 clusters



Conclusion:

- Customer segmentation is a highly effective strategy for organizations because it allows them to know which customers care about them and understand their needs enough to send a message that ensures brand success.
- we used RFM Modeling to see the relation between Recency,Frequncy and Monetary.
- After RFM model we used this data to perform clustering with the help of k mean clustering Algorithm.
- At the end we make 4 clusters of customers named as.
- **Clustor 0** - New Customer = Customer who transacted recently and have lower purchase frequency, with low amount of monetary spending.
- **Clustor 1** - Lost Customers = Customers with the least Monetary spending and the least number of transaction.
- **Clustor 2** - Best Customers = Most frequent spenders with the highest monetory spending amount and had transacted recently.
- **Clustor 3** - At Risk Customers = Customers who made their last transaction a while ago and made less frequent and low monetary purchases.

THANK YOU