Customer Segmentation

RFM MODELING AND K-MEANS CLUSTERING
APPLIED ON E-COMMERCE DATA

TEAM MEMBERS

Swati Basu
06004092020

Hritika Verma
01204092020

Sangeetha Panicker
04804092020

Overview of the Project

Imagine that you are treating the grocery shop owner that you shop every day, as you treat your significant other. That can be fun at the beginning, however may cause disastrous situations too. Likewise, it can be unfavourable for a company to manage its relationships with every customer similarly.

Companies that deploy customer segmentation are under the notion that every customer has different requirements and require a specific marketing effort to address them appropriately. Companies aim to gain a deeper approach of the customer they are targeting. Therefore, their aim has to be specific and should be tailored to address the requirements of each and every individual customer. Furthermore, through the data collected, companies can gain a deeper understanding of customer preferences as well as the requirements for discovering valuable segments that would reap them maximum profit. This way, they can strategize their marketing techniques more efficiently and minimize the possibility of risk to their investment.

Dataset

https://drive.google.com/drive/u/0/folders/1Y3NxA0-b3kZoDdfRSRN9g5MH-xc7rKne

Size : (541909* 8)

Source : Kaggle (E-Commerce Dataset)

Use Case

What we can achieve !

Customer segmentation enables a company to customize its relationships with the customers, as we do in our daily lives.

When you perform customer segmentation, you find similar characteristics in each customer’s behaviour and needs. Then, those are generalized into groups to satisfy demands with various strategies. Moreover, those strategies can be an input of the

- Targeted marketing activities to specific groups
- Launch of features aligning with the customer demand
- Development of the product roadmap

We can apply K-Means clustering after RFM modeling to the dataset and Data visualization can be very useful to show the required results.

Project Timeline

01

Phase 01

Topic Finalization
Dataset Explore
Data Analysis.

02

Phase 02

Data
Preprocessing

03

Phase 03

Feature
Engineering
Data Visualization

04

Phase 04

RFM Modeling

05

Phase 05

PCA Analysis
K-means
CLustering

06

Phase 06

Different
Clustering
Algorithms &
Accuracy
Performance

07

Phase 07

Reseach Paper

Individual Effort

Hritika Verma

Data Pre-Processing
Feature Engineering
Data Visualization
Presentation Designing and Content
Implemented Treemap after Clustering
Research Paper LaTex Writing
(Dataset Description,
Data Preprocessing,
Feature Engineering,
Data Visualization,
Future work and Conclusion,
Interpretation of results of all algorithms applied)

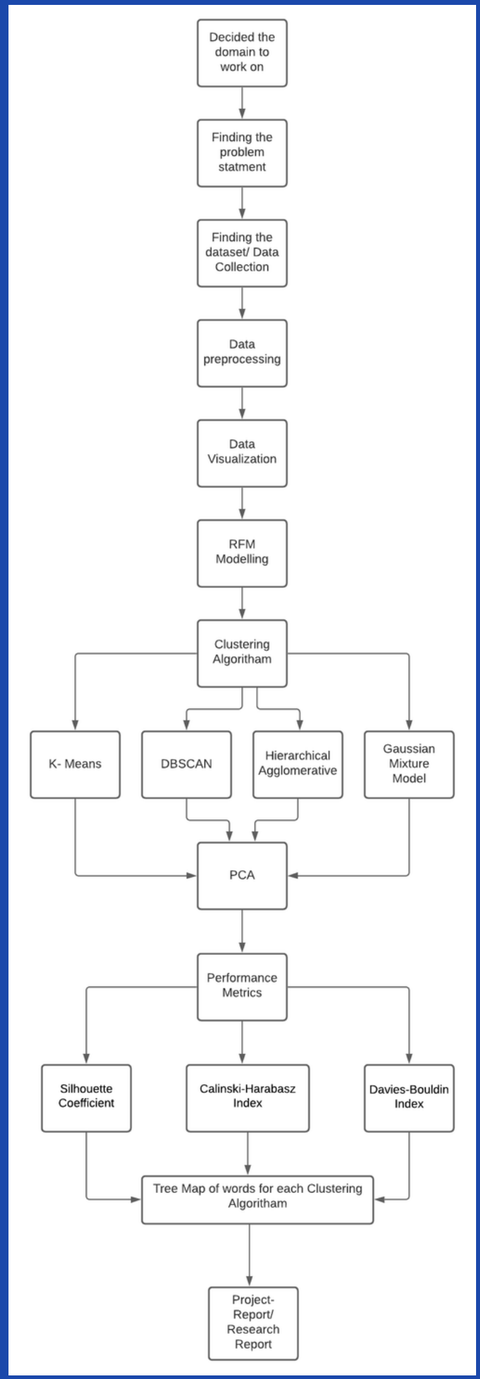
Sangeetha Panicker

RFM Modeling
and Visualization
Research Paper Work
PPT Content Modification
Research paper:
Abstract, Introduction,
Proposed Approach,
Complete theory of RFM modelling, Results and observation table for RFM,
Made interpretation table

Swati Basu

K-Means Clustering
PCA Analysis
DBSCAN Clustering
Hierarchical Clustering
Gaussian Mixture Model
Accuracy Performance - (Silhouette Coefficient, Calinski-Harabasz Index, Davies-Bouldin Index
Research Paper Writing
(SECTION - 3 : Work Flow Diag + Complete theories of clustering algo, PCA & Performance Metrics
SECTION - 4 : Complete Results and observations of all clustering algo
RESULT Interpretation of all Clustering Algortihms
PPT Designing and Content Modification

Work Flow Diagram



Pre-processing & Feature Engineering

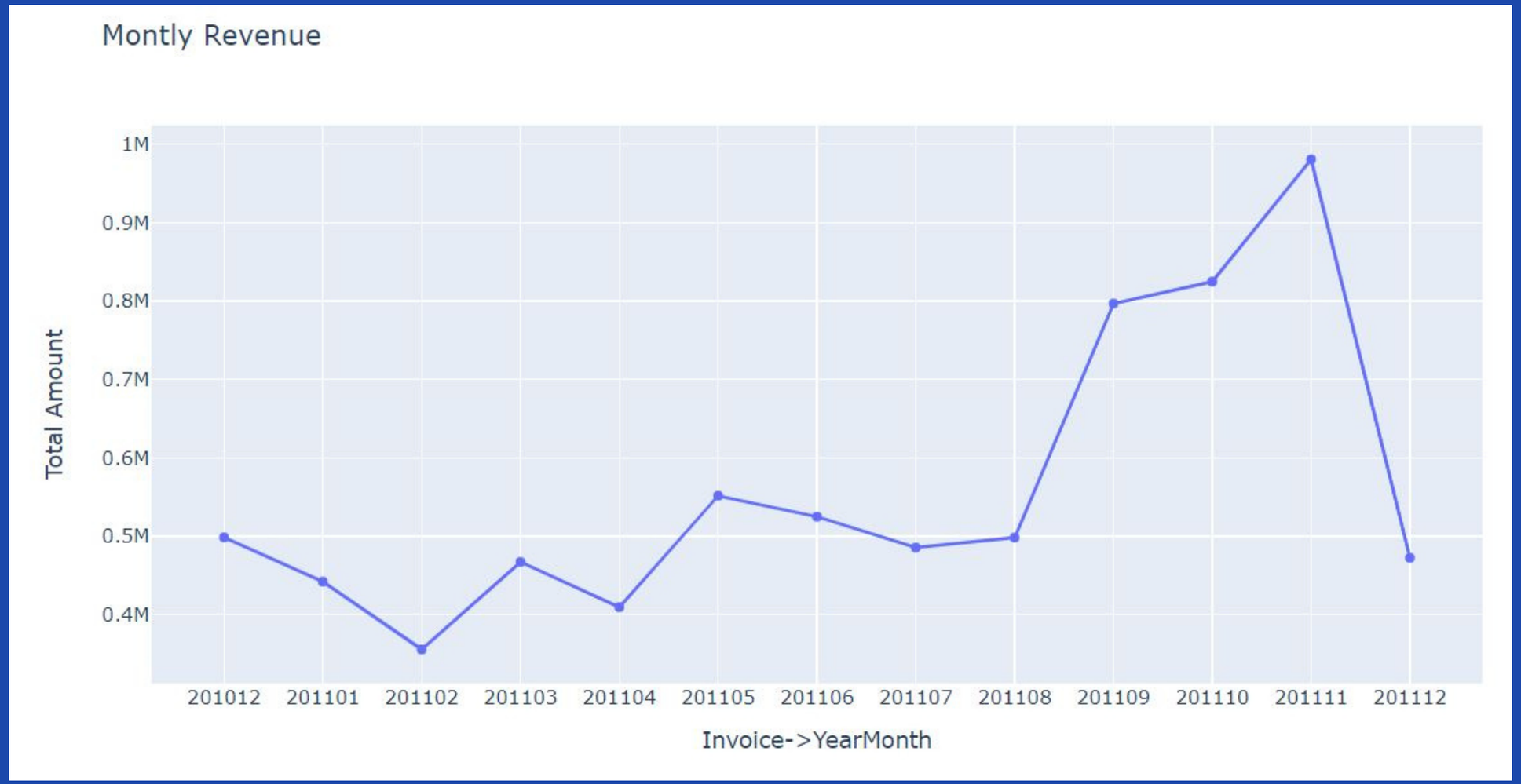
In any Machine Learning process, **Data Preprocessing** is that step in which the data gets transformed, or Encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm.

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.

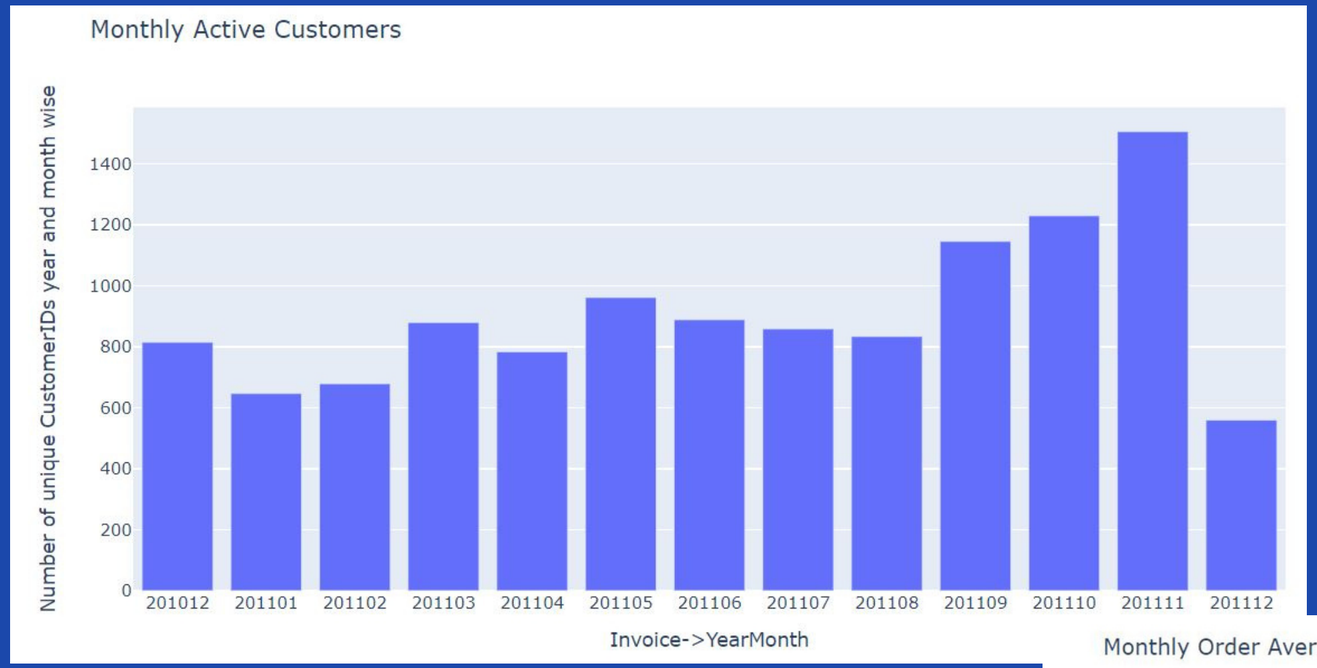
Following Features are the outcome of which we performed:

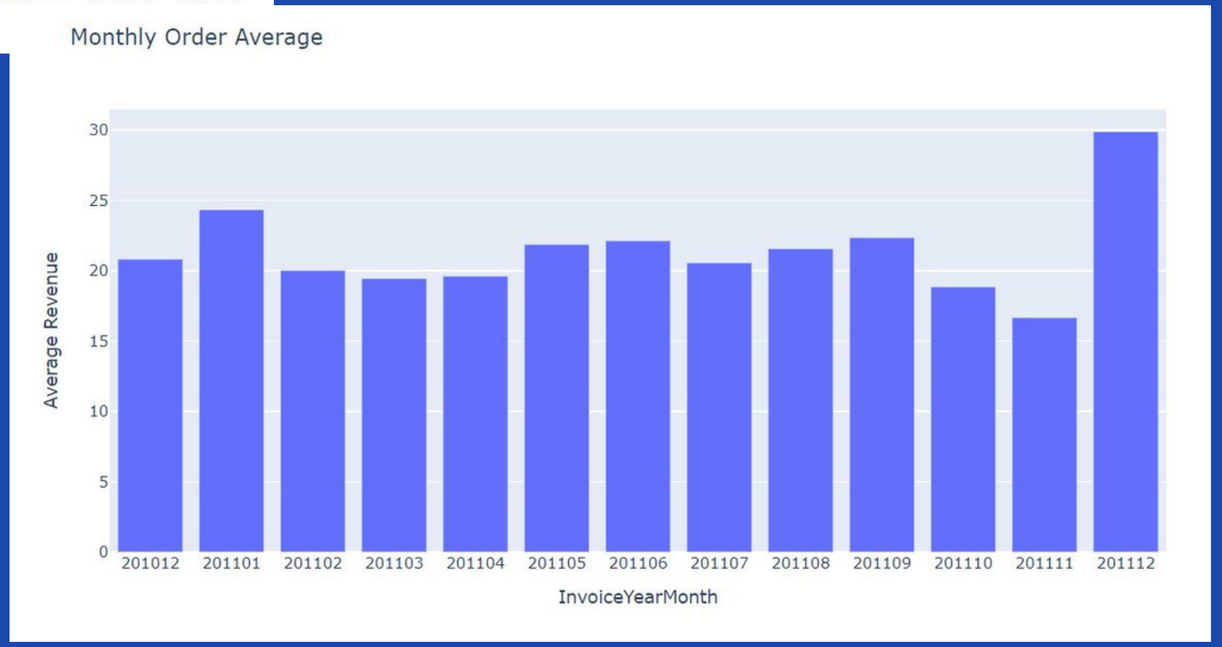
1. Monthly Revenue
2. Monthly Revenue Growth Rate
3. Monthly Active Customers
4. Monthly Order Count
5. Monthly Revenue per Order
6. New Customer Ratio

Visualization Result



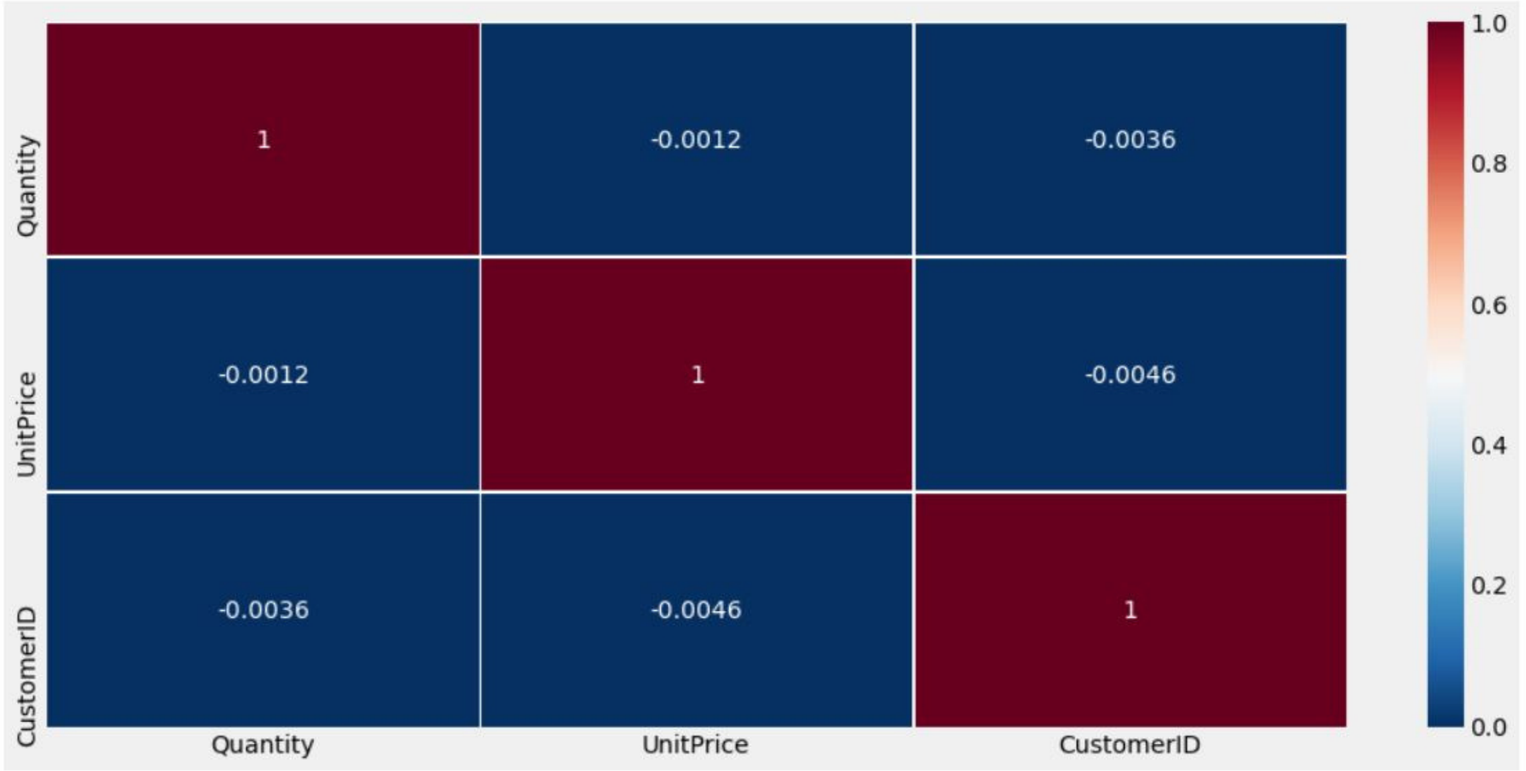
Visualization Result





Visualization Result

Visualization Result



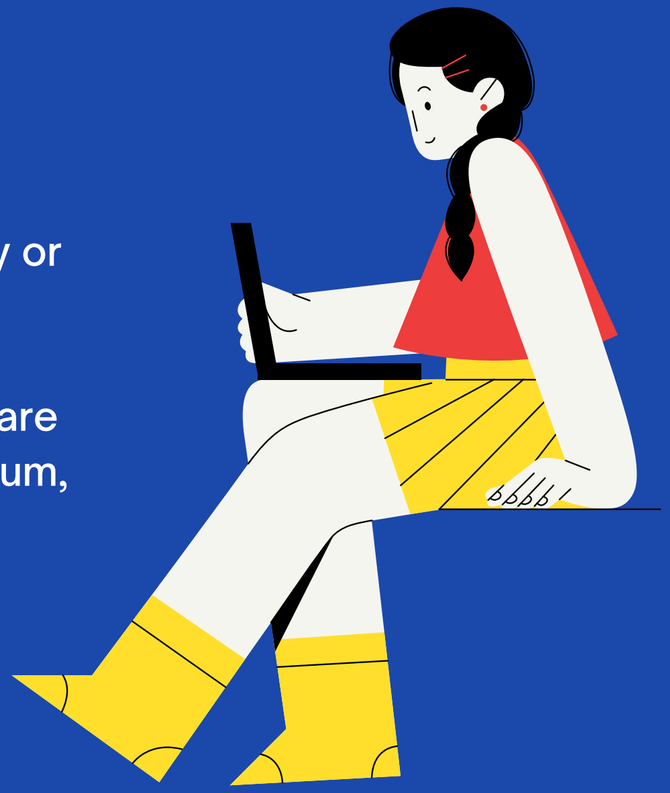
Since, It is an Unsupervised learning project i.e. Customer Segmentation, thats why there's no correlation between the attributes of data as all the values are in negative state.

RFM Modeling

RFM analysis is a data-driven customer behavior segmentation technique.
RFM stands for recency, frequency, and monetary value.
The idea is to segment customers based on when their last purchase was, how often they've purchased in the past, and how much they've spent overall.
Recency: Date of Last of Purchase
Frequency: Total Number of Orders
Monetary: Average Order Value

To calculate the scores of each customer based on recency, frequency, and monetary or our FM modeling to create several clusters of customers according to their spending behavior, recency purchases as well as how frequently they are buying we would also look at how to create clusters of most loyal customers as well as the customers who are on the verge of churning out that is grouping them into various loyalty levels like platinum, gold, silver, and bronze where

- Platinum group represents the most loyal customers
- Gold: Are recent customers and spends more than silver but not frequently.
- Silver is those are Spends less than the gold and not much frequent to visit the platform.
- The bronze group represents the group that has not purchased anything for quite long.

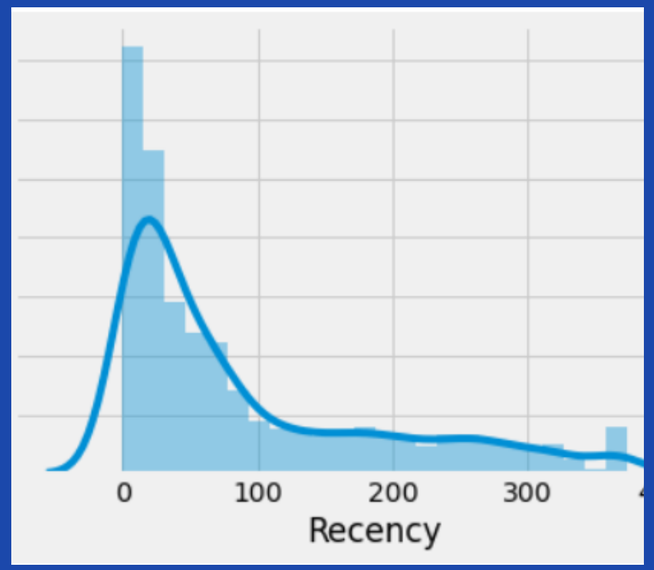


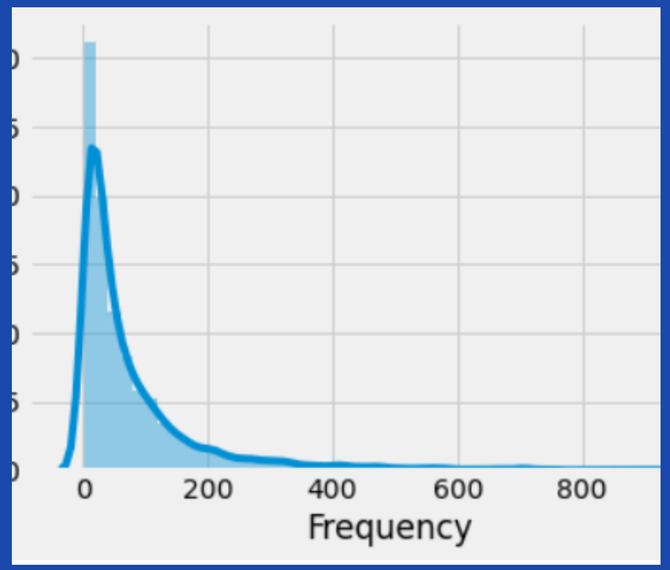
RFM Modeling

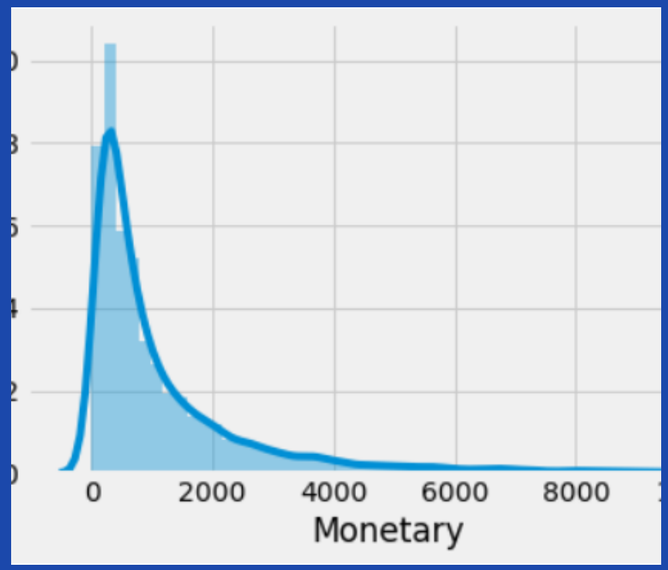
Recency = Latest Date - Last Inovice Data,
Frequency = count of invoice no. of the transaction(s)
Monetary = Sum of Total Amount for each customer

- More the recent purchase more will be the recency score
- How frequently the customer visits the site, more will be the frequency.
- The more the spending, more will be the monetary score.

These are the distribution plots of recency, frequency and monetary.





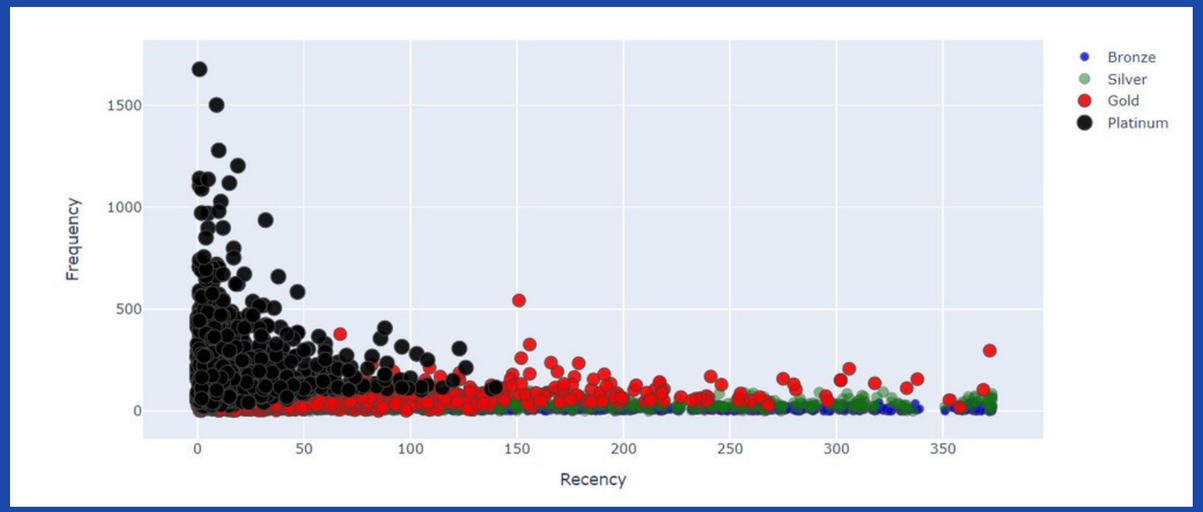


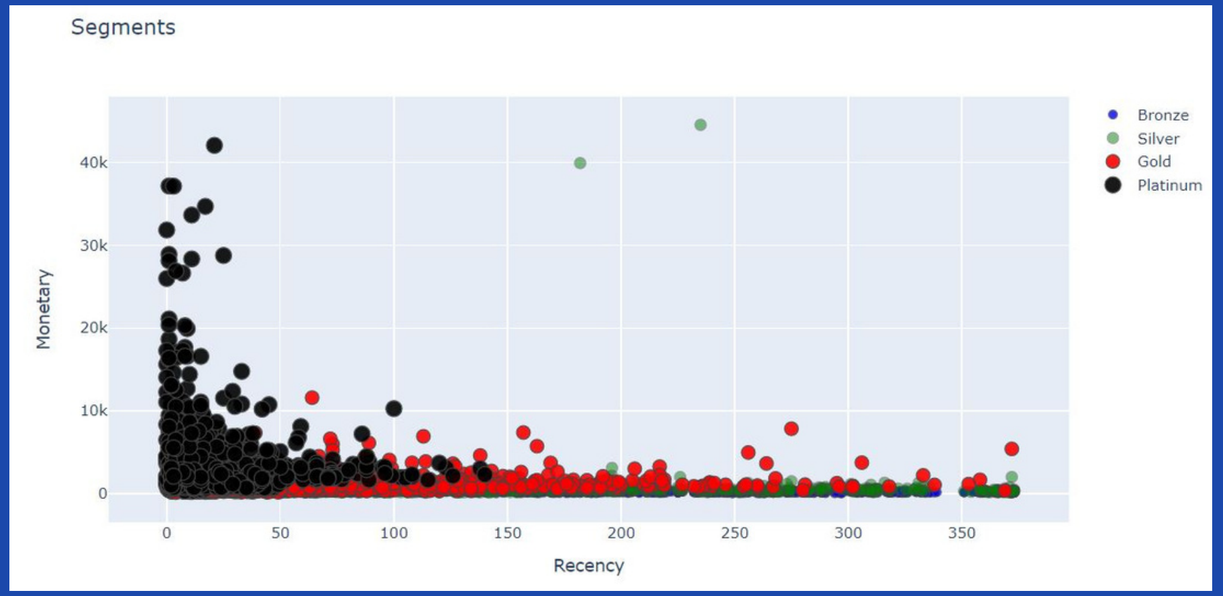
Ranking the customer on basis of RFM Scores

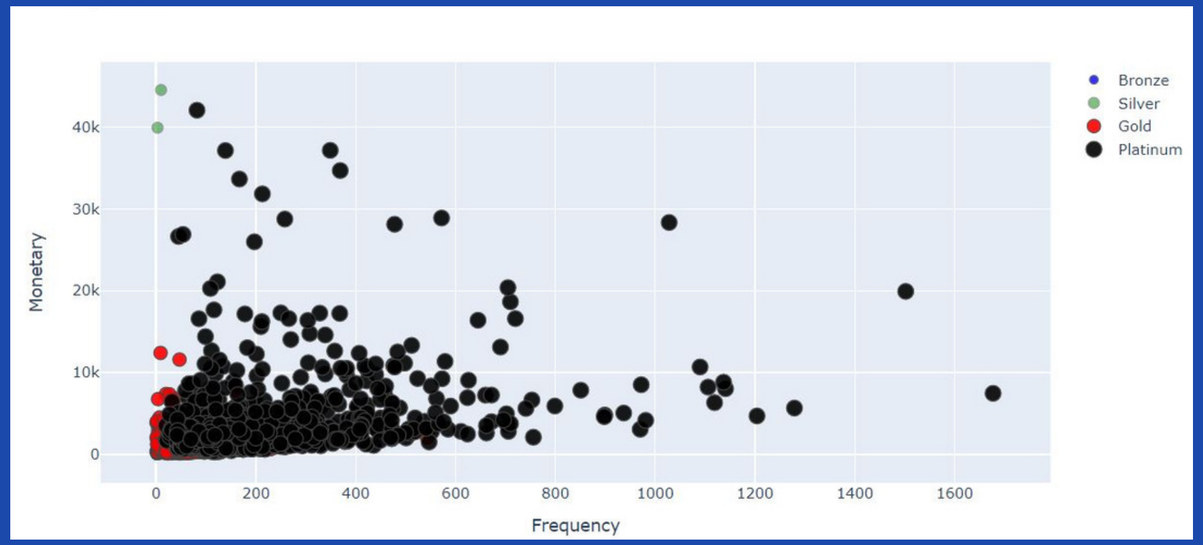
Loyality	RFM Score		
Platinum	3	5	4
Gold	7	8	6
Silver	10	9	
Bronze	11	12	

- The simple approach of scaling customers from 1-4 will result in different RFM scores, ranging from 444(lowest) to 111(highest).
- Each RFM cell will differ in size and vary from one another, in terms of the customer's key habits, captured in the RFM score.
- The customer with lowest RFM score will be most loyal customer.

RFM Modeling Result







K-Means Clustering

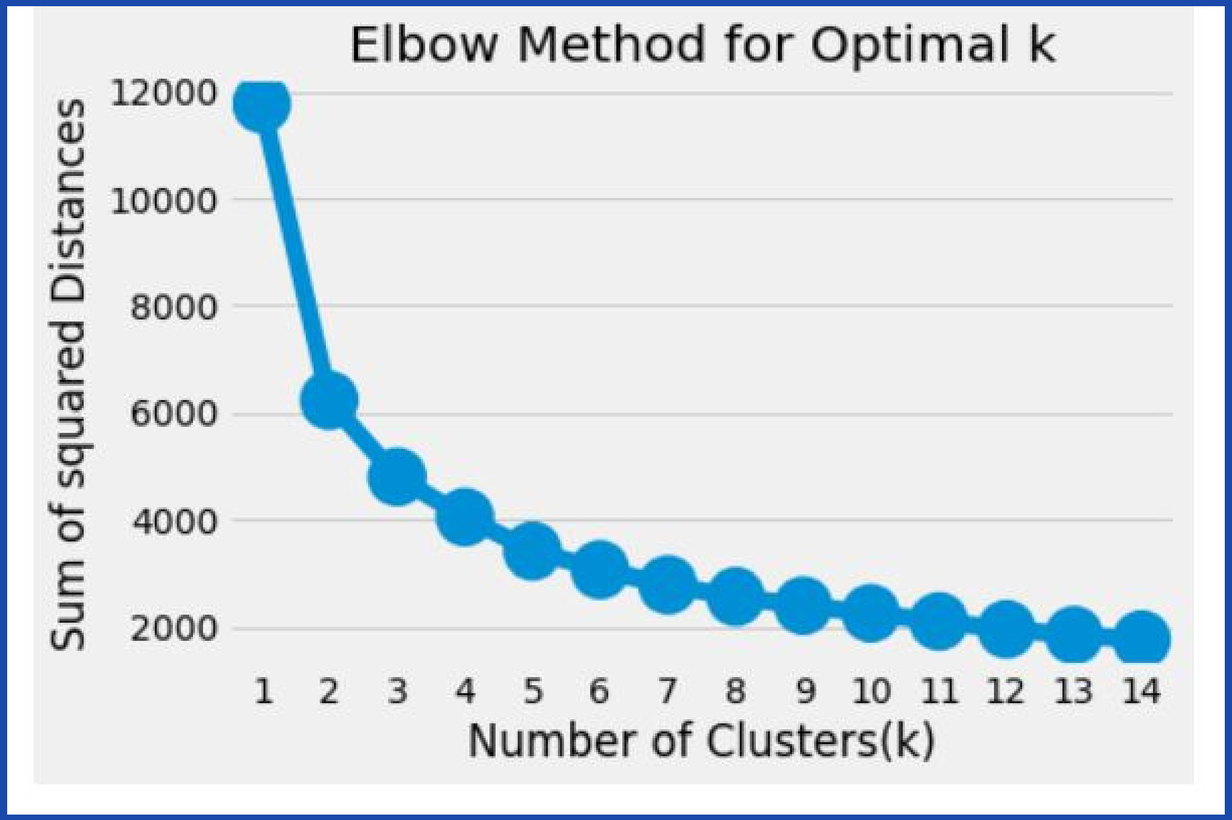
K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.
Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.

-->We implemented K-means Clustering on the basis of RFM modeling we did and found 4 types of cluster of customers i.e. Bronze,Silver,Gold and Platinum.
-->We observed that through K-Means clustering we got 3 clusters :-

Gold = Cluster 0
Silver + Bronze = Cluster 1
Platinum = Cluster 2

So, K-means was better as it computed with less no. of clusters , due to which, customer segmentation become easier for any sales company should refer.

K-Means Clustering Result



Elbow Method - Running the algorithm multiple times over a loop with an increasing no. of clusters choice and then plotting a clustering score as a function of no. of clusters when K increases the centroids are closer to cluster points.

--> Improvements will decline at some point rapidly giving the elbow shape

RESULT -> The curve dramatically decreases at 3 giving the optimal value for K.

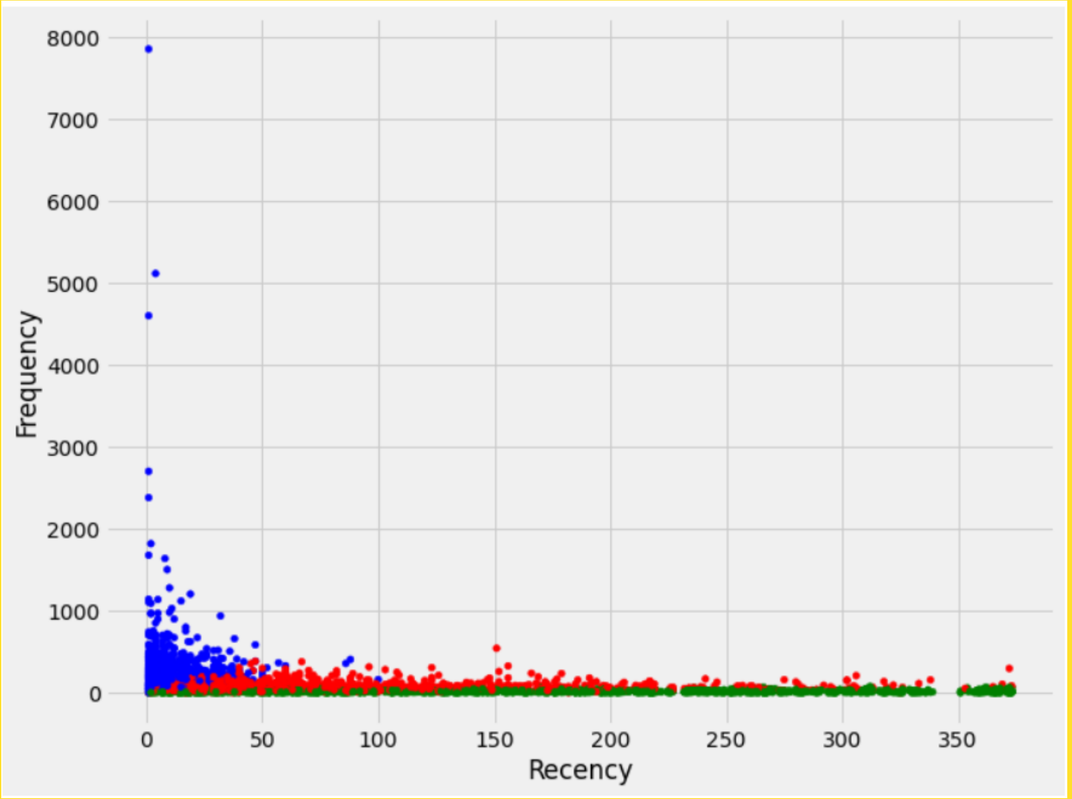
K = 3

K-Means Clustering Result

Cluster 0 --> Red (Gold)
->Spends less than the platinum group and not much frequent to visit the platform.

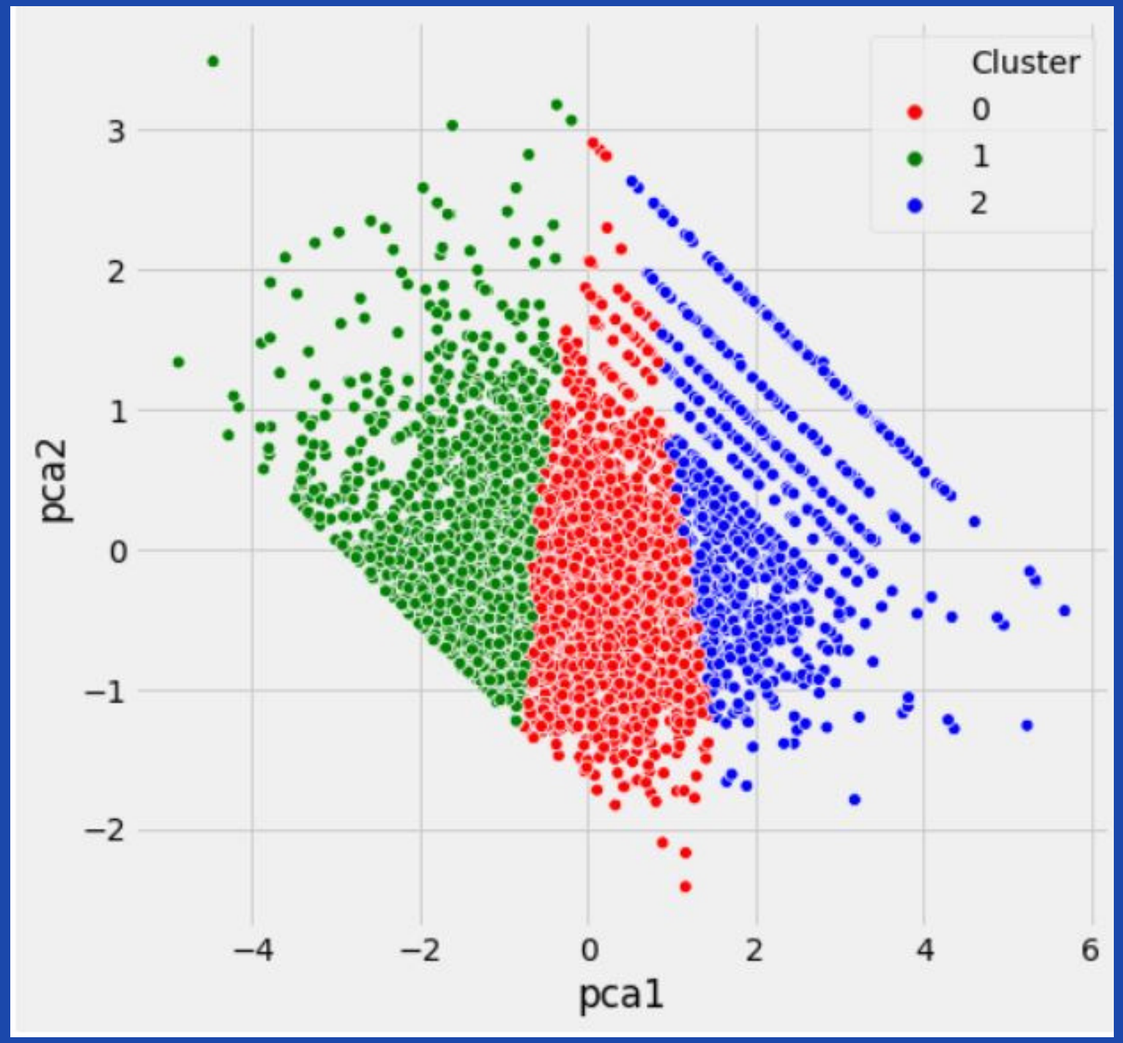
Cluster 1 --> Green (Silver + Bronze)
-> One who hasn’t purchased from the brand from quite long and he or she may be on the verge of churning out.

Cluster 2 --> Blue (Platinum)
->The most valuable and loyal customer that they don't want to lose.



Clusters	Platinum	Gold	Silver	Bronze
Cluster 0	328	1106	230	0
Cluster 1	0	36	665	705
Cluster 2	812	39	0	0

PCA Result



The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent.

DIMENSIONS = 2

DBSCAN Clustering Algorithm

Density-Based Clustering refers to unsupervised learning methods that identify distinctive groups/clusters in the data, based on the idea that a cluster in data space is a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a base algorithm for density-based clustering. It can discover clusters of different shapes and sizes from a large amount of data, which is containing noise and outliers.

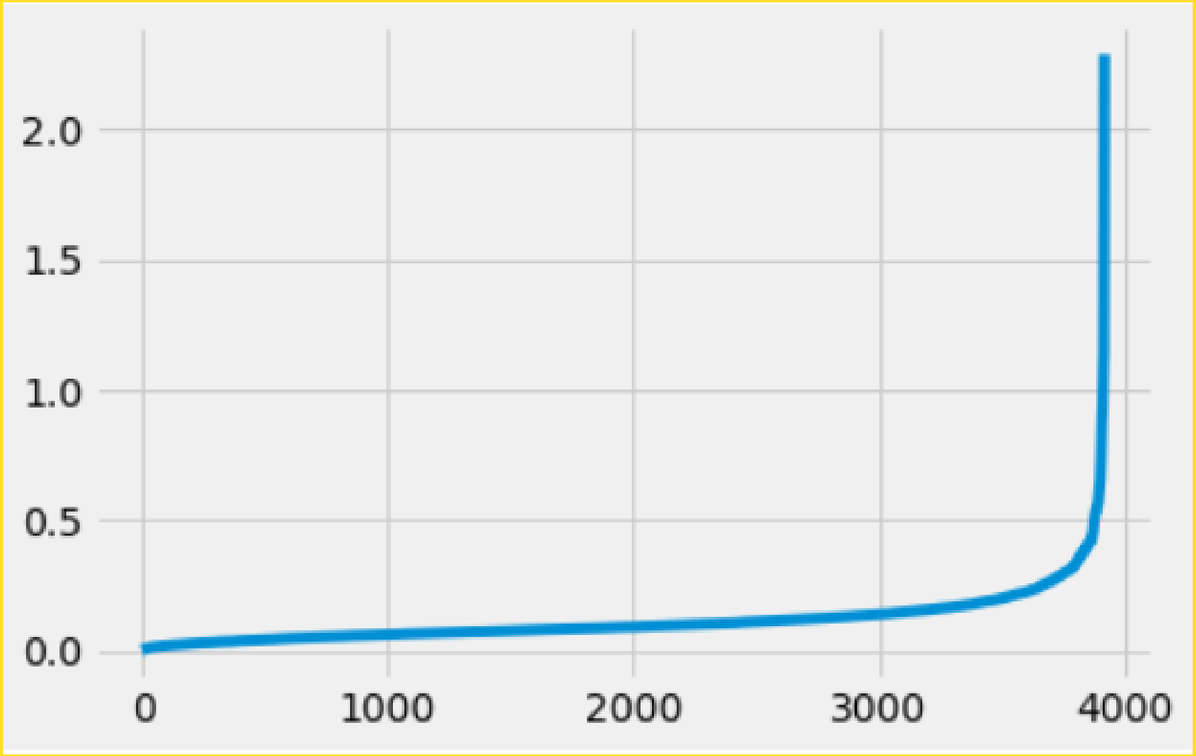
The DBSCAN algorithm uses two parameters:

- **minPts:** The minimum number of points (a threshold) clustered together for a region to be considered dense.
- **eps (ε):** A distance measure that will be used to locate the points in the neighborhood of any point.

DBSCAN Clustering Algorithm

- **MinPts** = 2*dim, where dim= the dimensions of data set (By Thumb Rule)

- **eps (ε) =** Calculated the average distance between each point and its k nearest neighbors, where k = the MinPts value we selected. The average k-distances are then plotted in ascending order on a k-distance graph. We find the optimal value for ε at the point of maximum curvature (i.e. where the graph has the greatest slope).



EPS = 0.4

DBSCAN Clustering Algorithm Result & Interpretation

Cluster 0 --> Orange

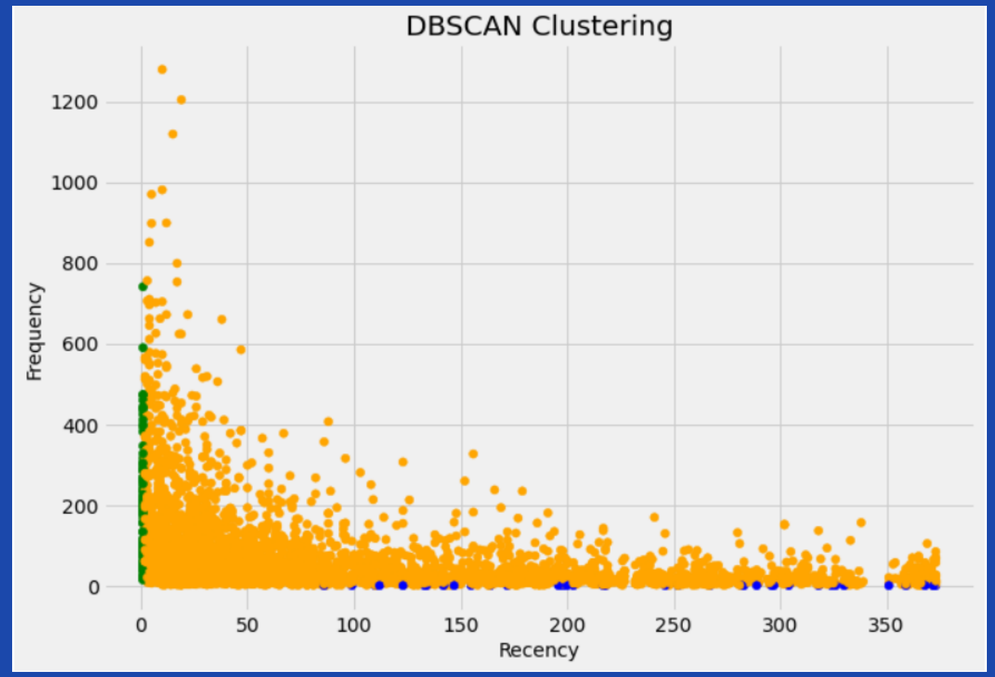
--> It includes most of those who are less loyal than platinum(most loyal) customers and less frequent in recent times along with most loyal customers.

Cluster 1 --> Green

--> It includes 90 perc. most loyal customers.

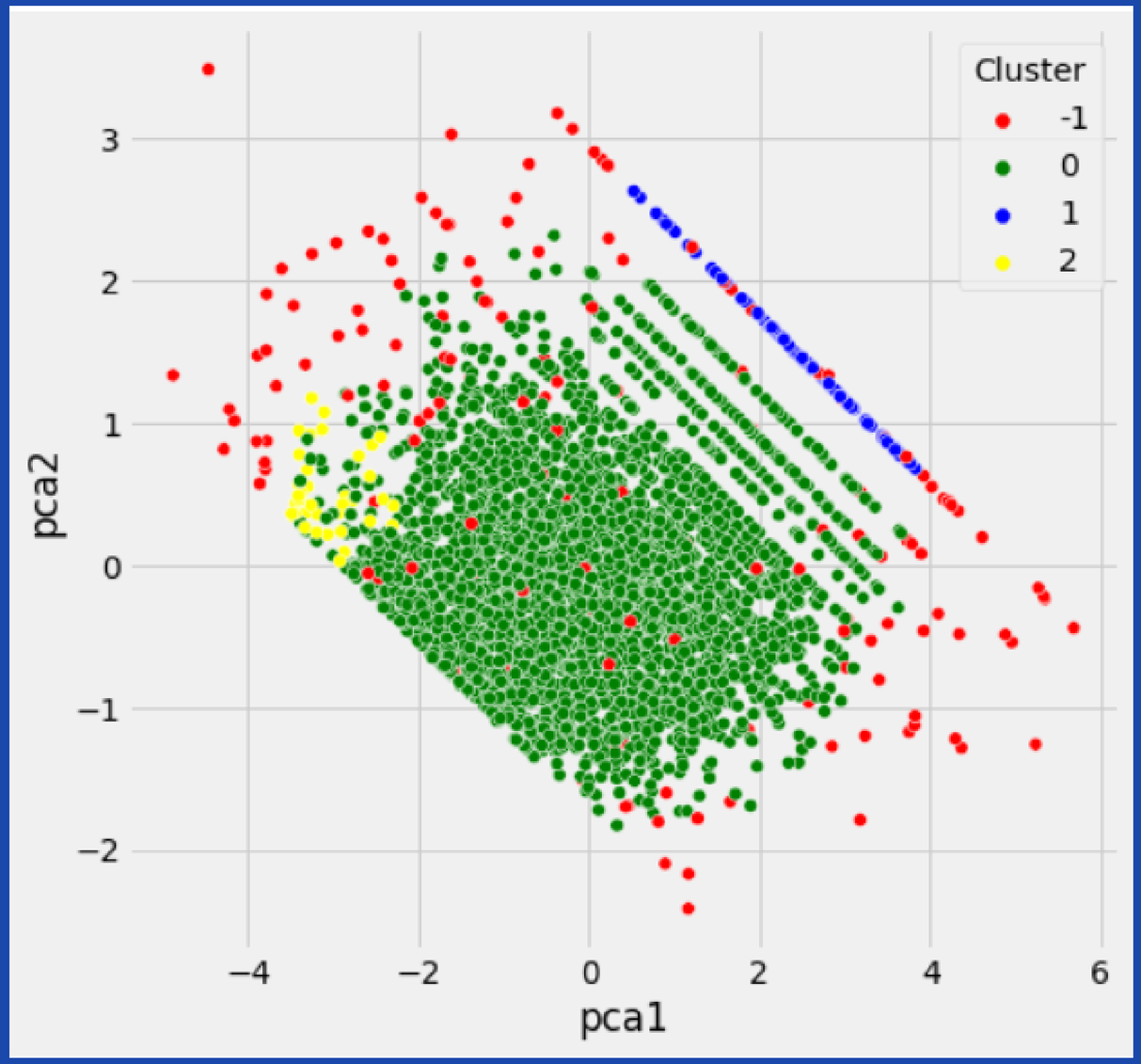
Cluster 2 --> Blue

--> It has bronze customers who are at the churning out phase.



Clusters	Platinum	Gold	Silver	Bronze
Cluster 0	995	1128	847	648
Cluster 1	87	10	0	0
Cluster 2	0	0	1	36

PCA applied to DBSCAN



Cluster --> - 1 indicated with Red colour represents the outliers.

Hierarchical Clustering

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

For e.g: All files and folders on our hard disk are organized in a hierarchy.

The algorithm groups similar objects into groups called clusters. The endpoint is a set of clusters or groups, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

This clustering technique is divided into two types:

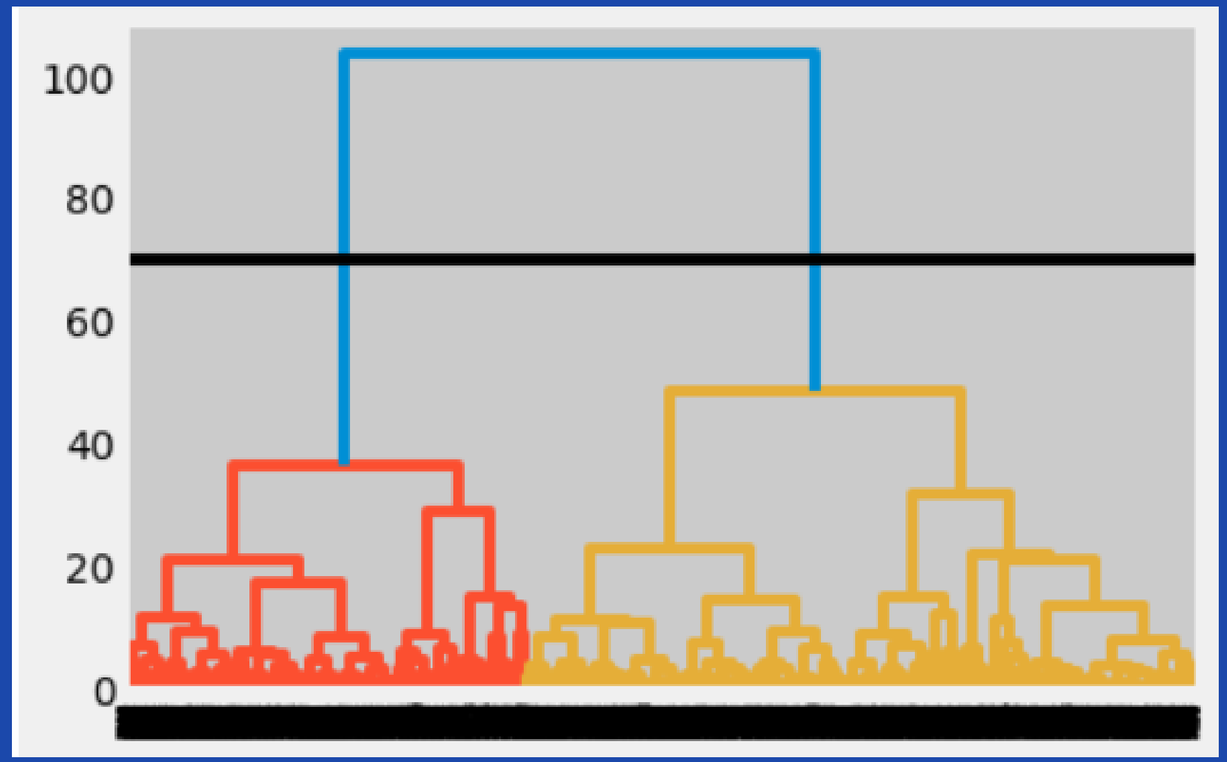
1. **Agglomerative Hierarchical Clustering**
2. **Divisive Hierarchical Clustering**

Agglomerative Hierarchical Clustering

- **Agglomerative** — Bottom up approach. Start with many small clusters and merge them together to create bigger clusters.

We can use a **DENDROGRAM** to visualize the history of groupings and figure out the optimal number of clusters.

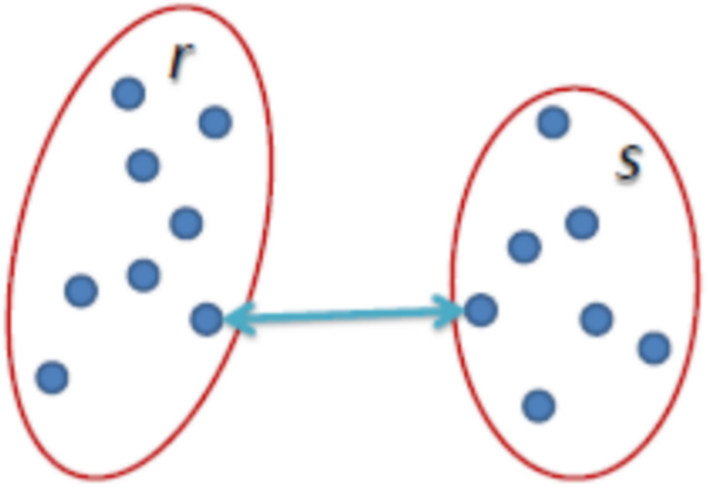
1. Determine the largest vertical distance that doesn't intersect any of the other clusters
2. Draw a horizontal line at both extremities
3. The optimal number of clusters is equal to the number of vertical lines going through the horizontal line



(LINKAGE = 'WARD')

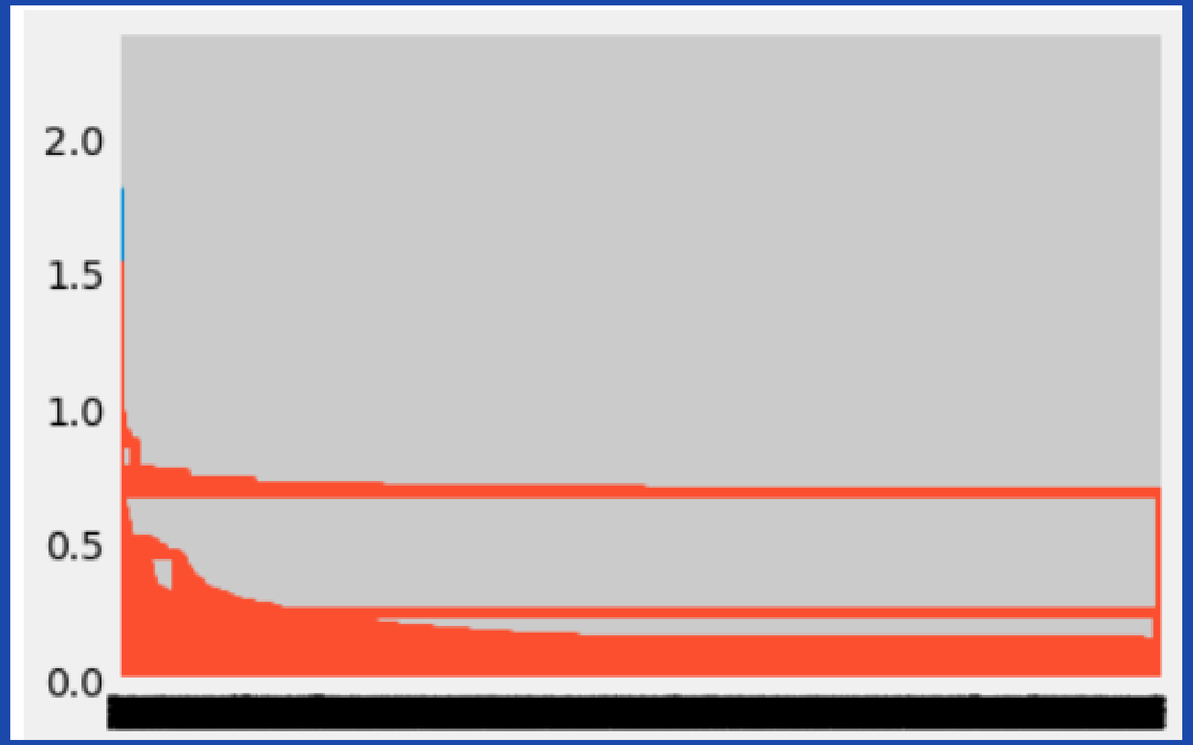
CLUSTERS = 2

Linkage Criteria

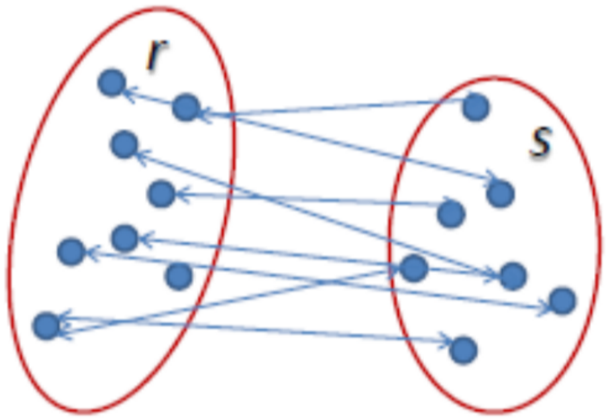


$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

1. **Single Linkage : -** The distance between two clusters is the shortest distance between two points in each cluster

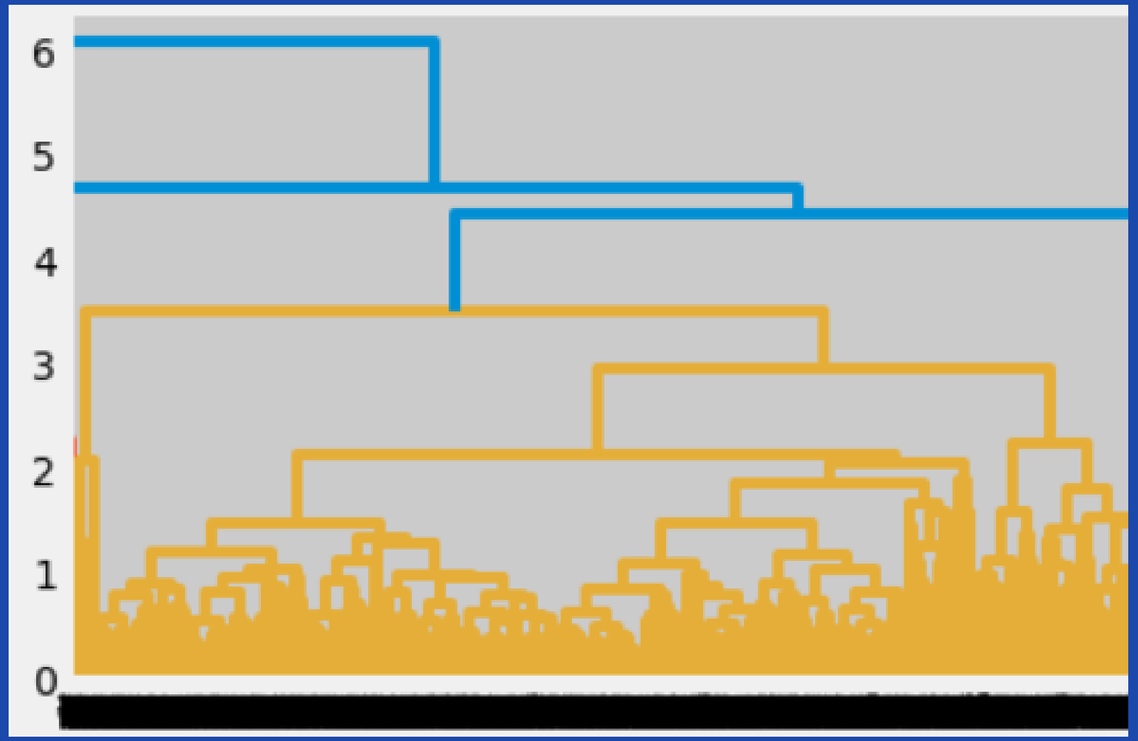


Linkage Criteria

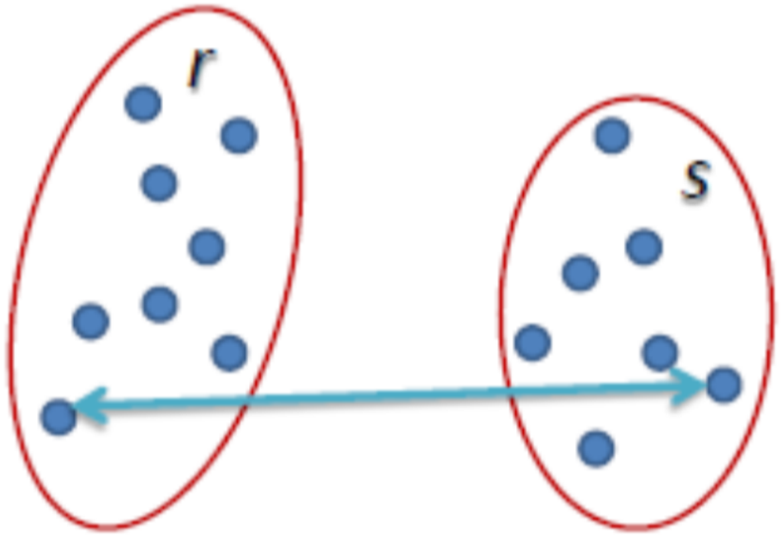


$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

1. **Average Linkage : -** The distance between clusters is the average distance between each point in one cluster to every point in other cluster.

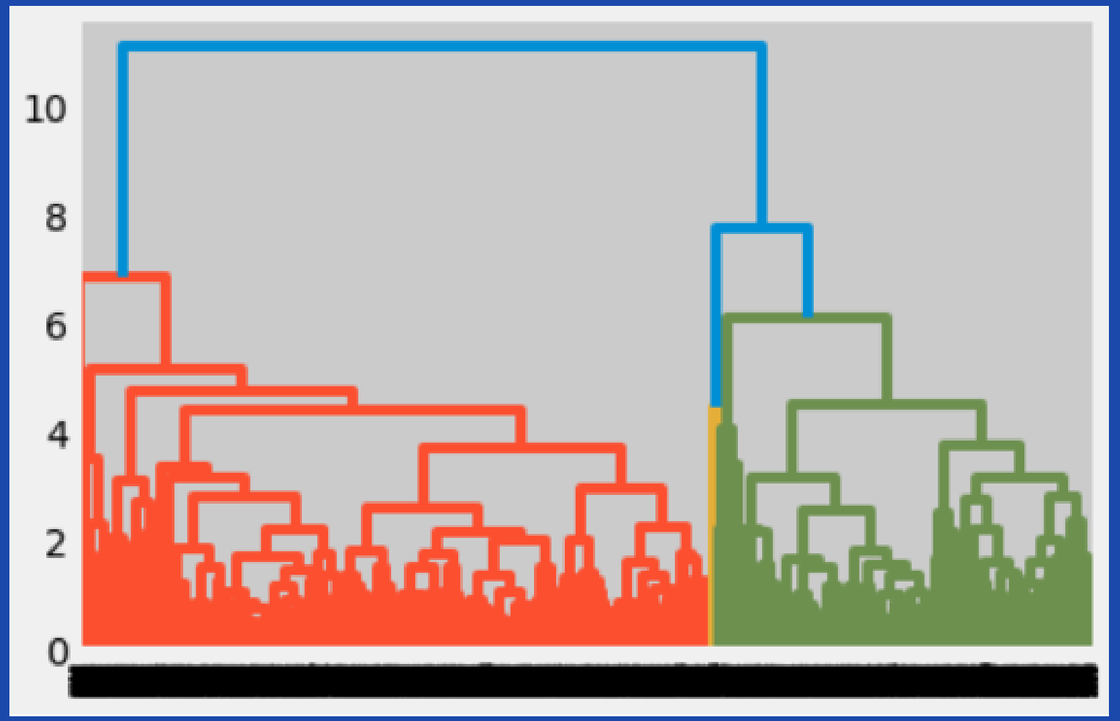


Linkage Criteria

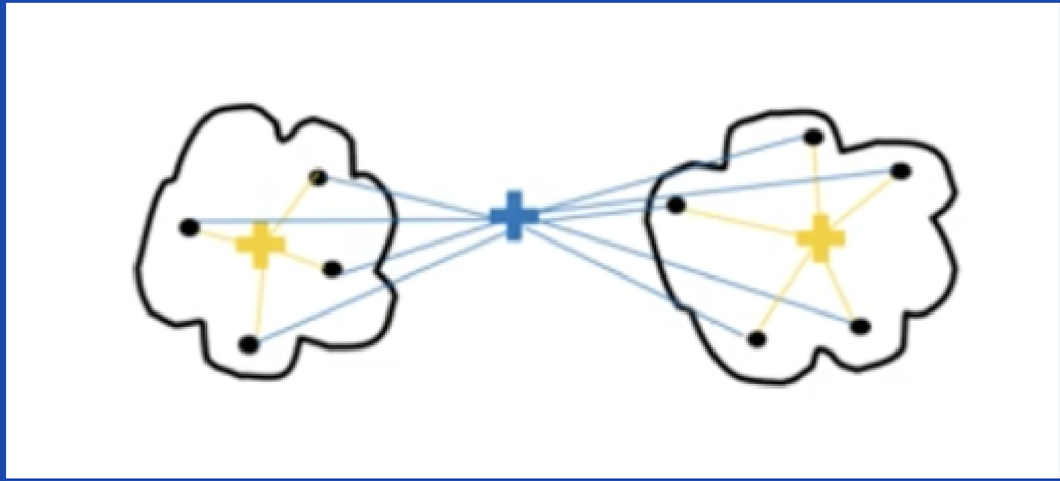


$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

2. Complete Linkage : - The distance between two clusters is the longest distance between two points in each cluster

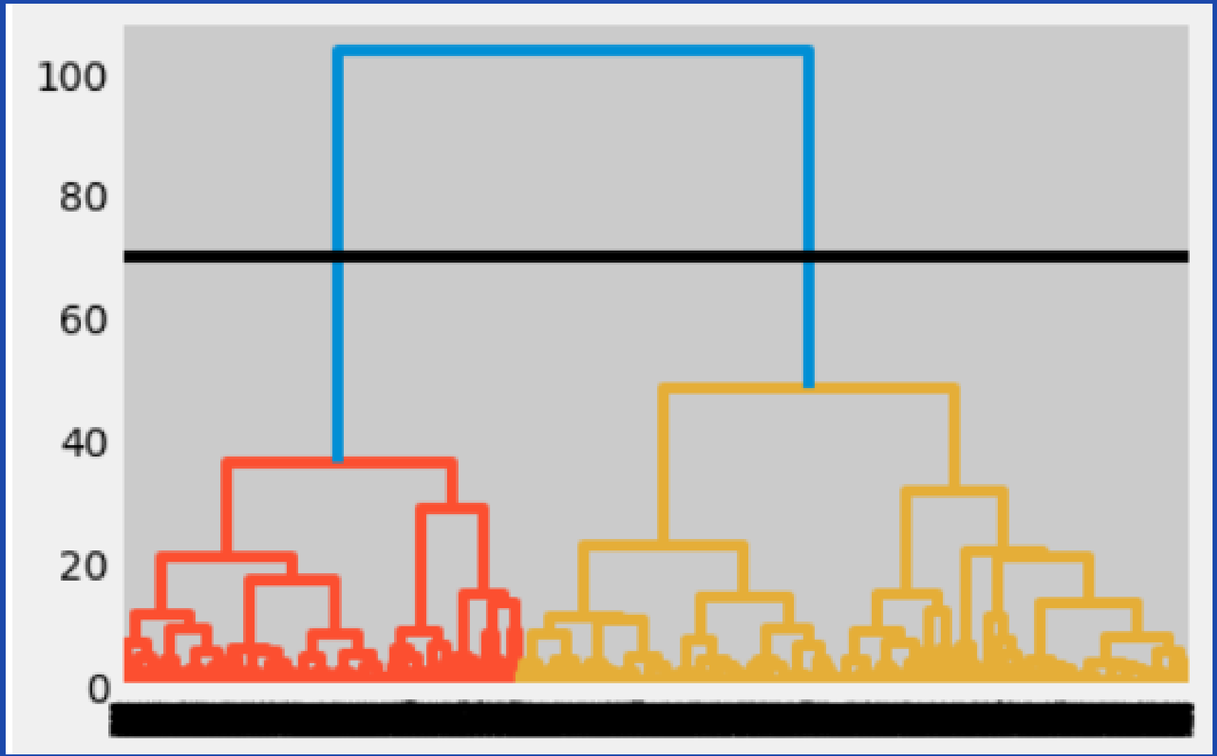


Linkage Criteria

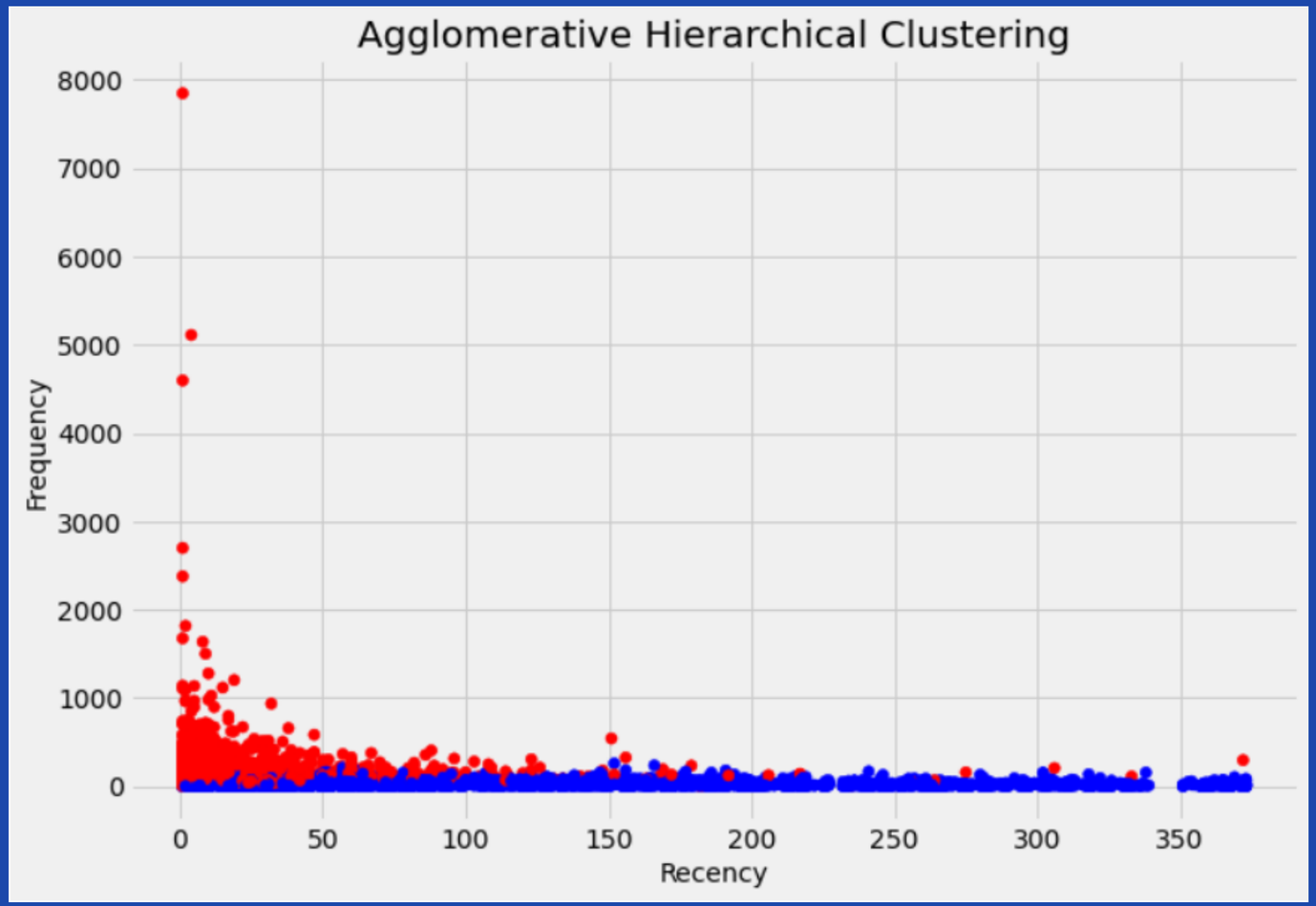


1. **Ward Linkage : -** The distance between clusters is the sum of squared differences within all clusters.

CLUSTERS = 2



Agglomerative Hierarchical Clustering

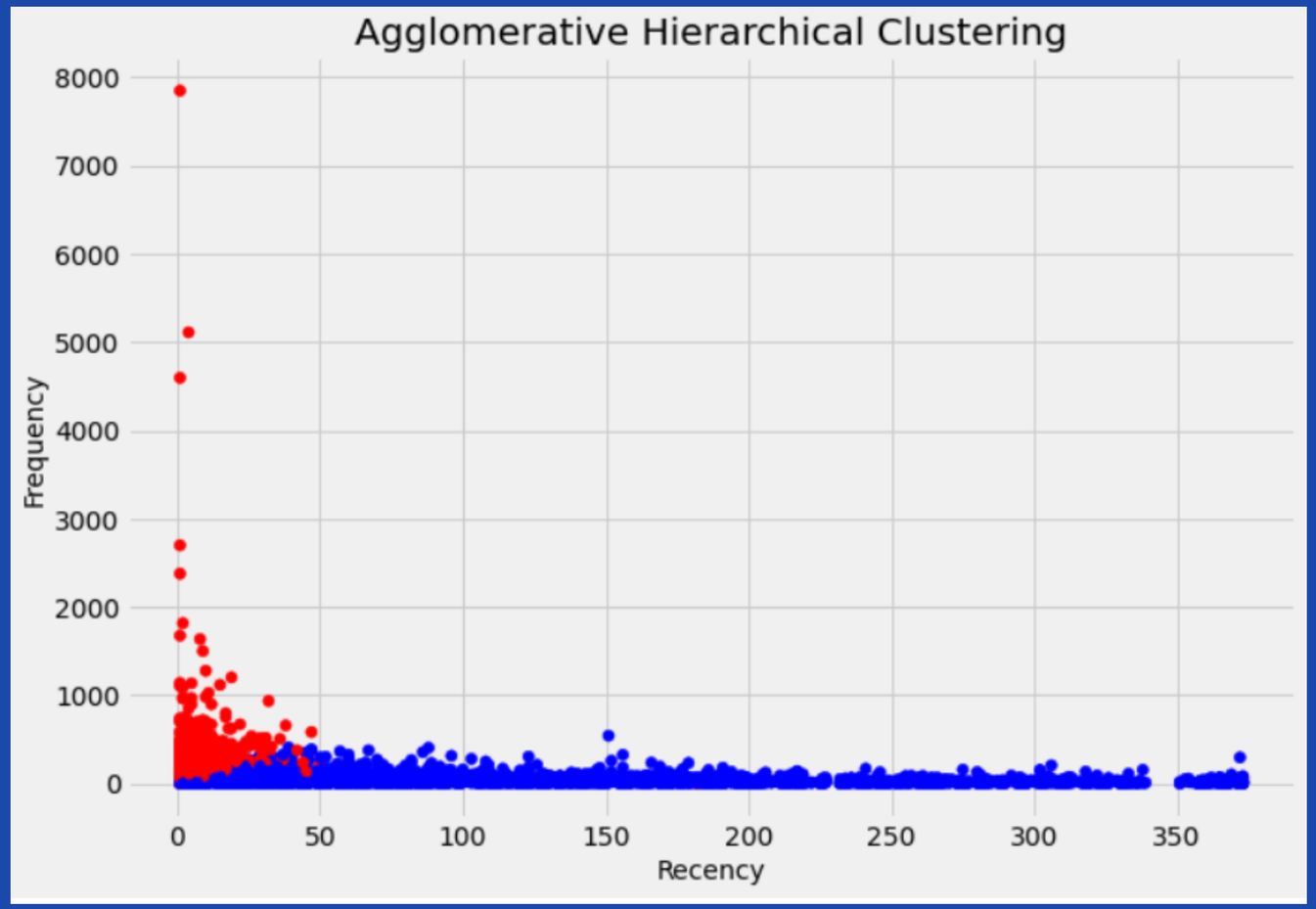


LINKAGE - WARD,

DISTANCE METRIC - EUCLIDEAN

CLUSTERS = 2

Agglomerative Hierarchical Clustering

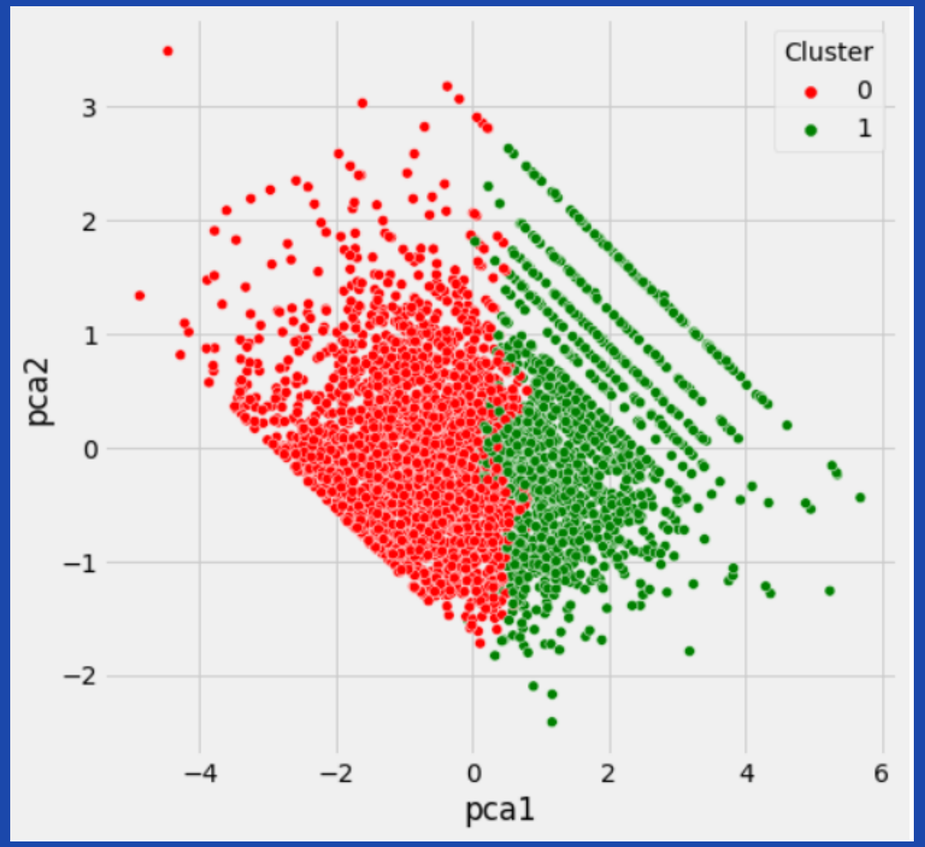


LINKAGE - COMPLETE,

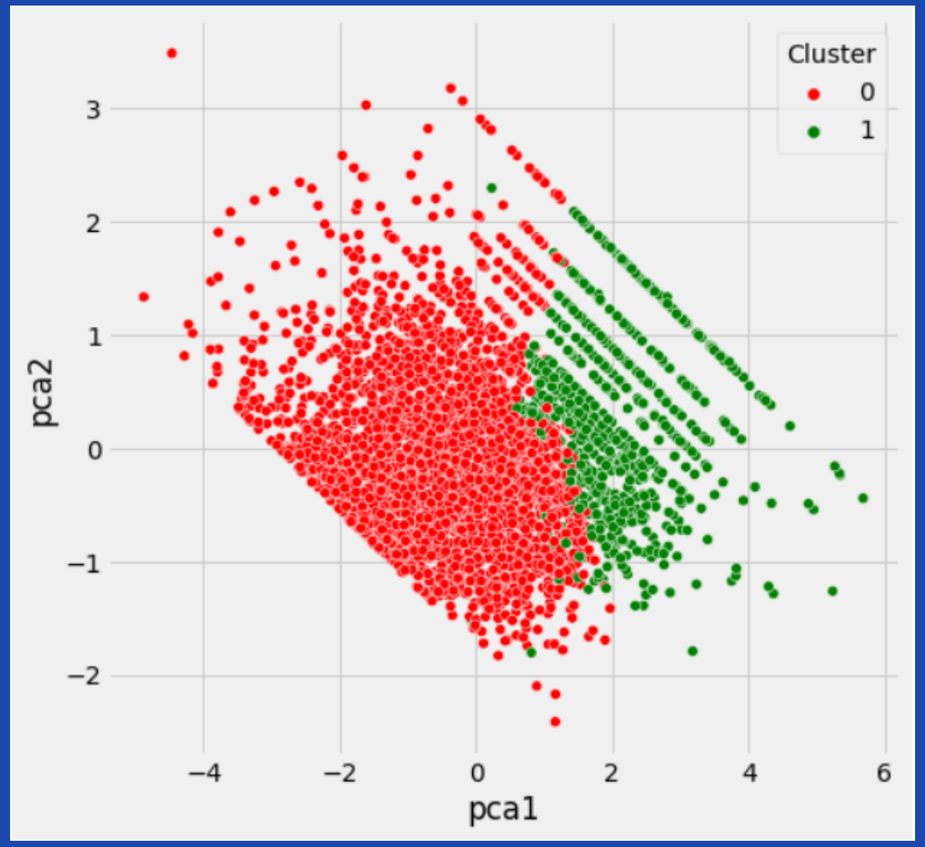
DISTANCE METRIC - MANHATTAN

CLUSTERS = 2

PCA applied on Agglomerative Hierarchical Clustering



LINKAGE - WARD, DISTANCE METRIC - EUCLIDEAN



LINKAGE - COMPLETE, DISTANCE METRIC - MANHATTAN

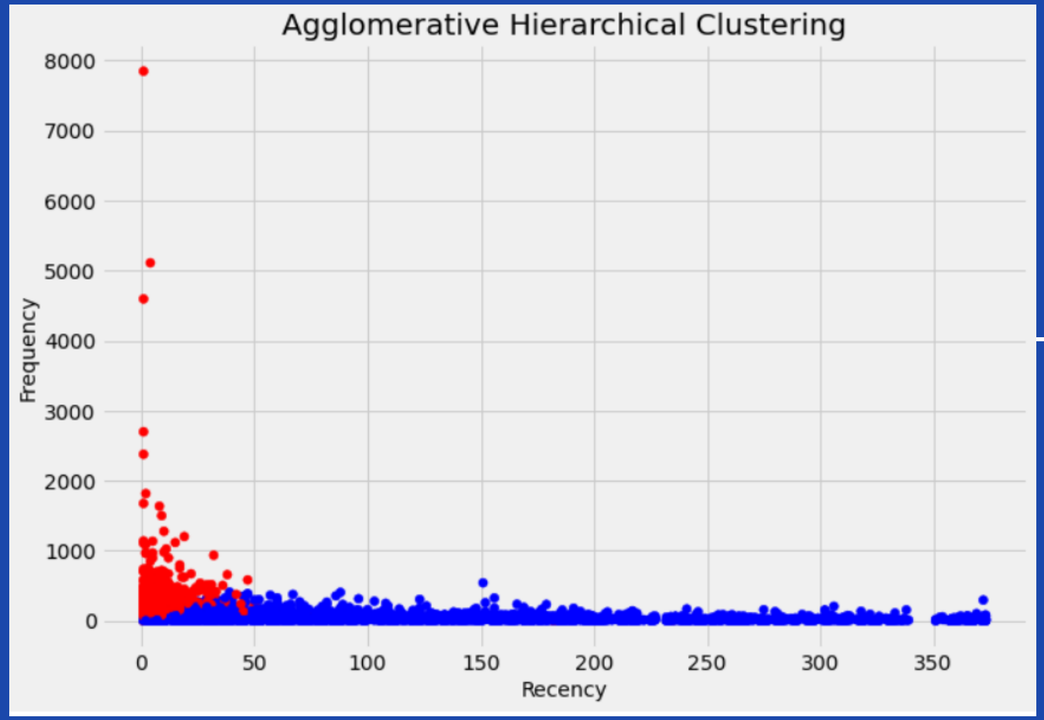
Agglomerative Hierarchical Clustering Result & Interpretation

Cluster 0 --> Red

--> It includes most of those who are least loyal customers(silver) and not frequent in recent times along with more loyal customers(gold) and those who are at churning out phase(bronze) or simply includes all types except the most loyal ones.

Cluster 1 -->Blue

--> It includes approx 80 perc. most loyal customers along with less loyal ones i.e. gold



Clusters	Platinum	Gold	Silver	Bronze
Cluster 0	26	833	892	705
Cluster 1	1114	348	3	0

Gaussian Mixture Model Selection

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.

The GaussianMixture object implements the expectation-maximization (EM) algorithm for fitting mixture-of-Gaussian models. It can also draw confidence ellipsoids for multivariate models, and compute the Bayesian Information Criterion to assess the number of clusters in the data.

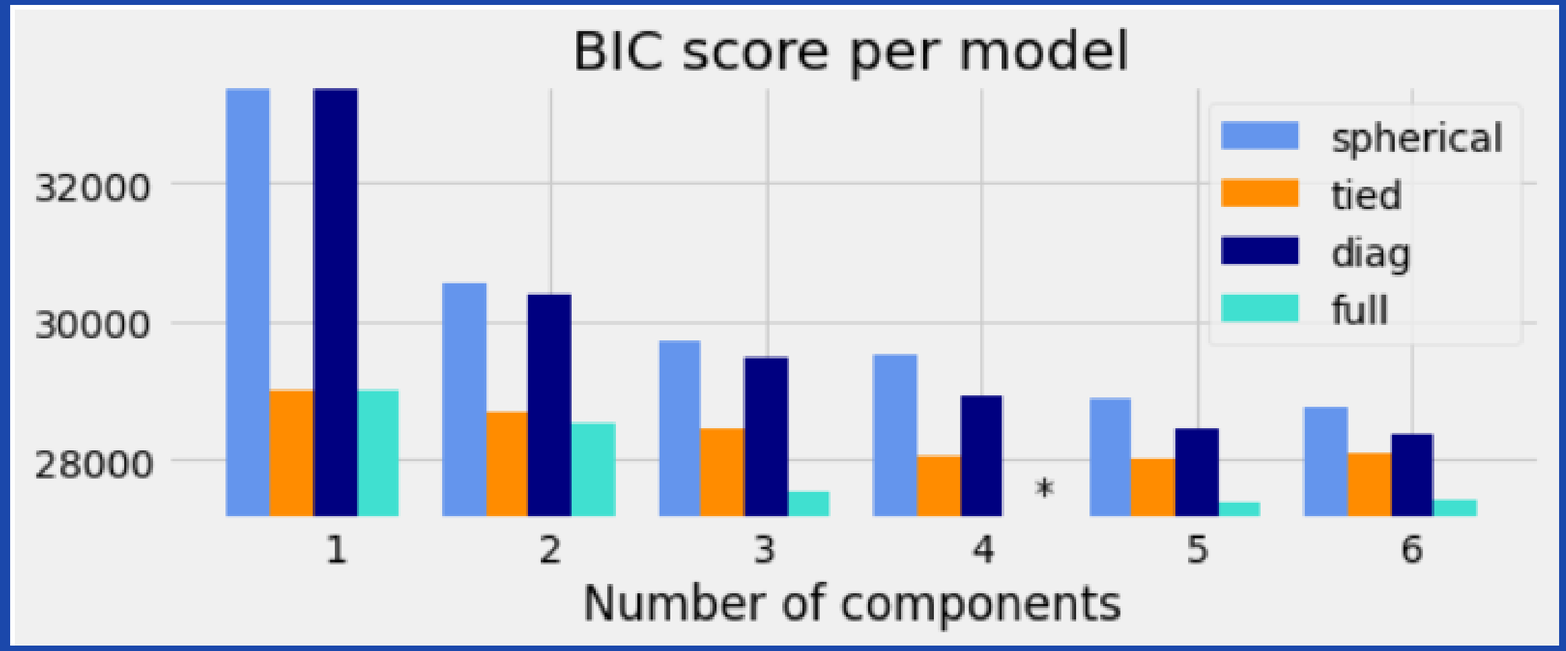
The GaussianMixture comes with different options to constrain the covariance of the difference classes estimated: spherical, diagonal, tied or full covariance.

Gaussian Mixture Model Selection

The BIC criterion can be used to select the number of components in a Gaussian Mixture in an efficient way. In theory, it recovers the true number of components only in the asymptotic regime (i.e. if much data is available and assuming that the data was actually generated i.i.d. from a mixture of Gaussian distribution).

CLUSTERS = 4

The model with the lowest BIC is selected



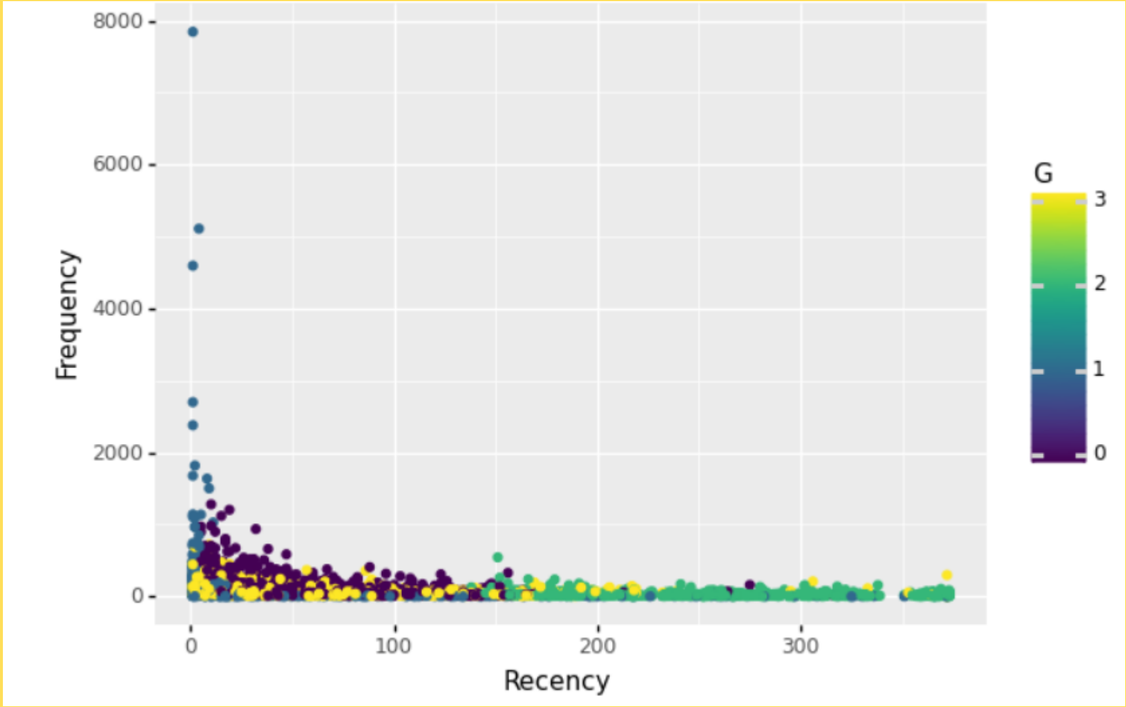
Gaussian Mixture Model Selection

Cluster 0 --> Purple
--> It includes most of those at churning out phase(bronze) or simply includes all types except the most loy

Cluster 1 --> Blue
--> It includes approx 80 perc. most loyal customers along with less loyal ones i.e. gold and very less number of silver and bronze types.

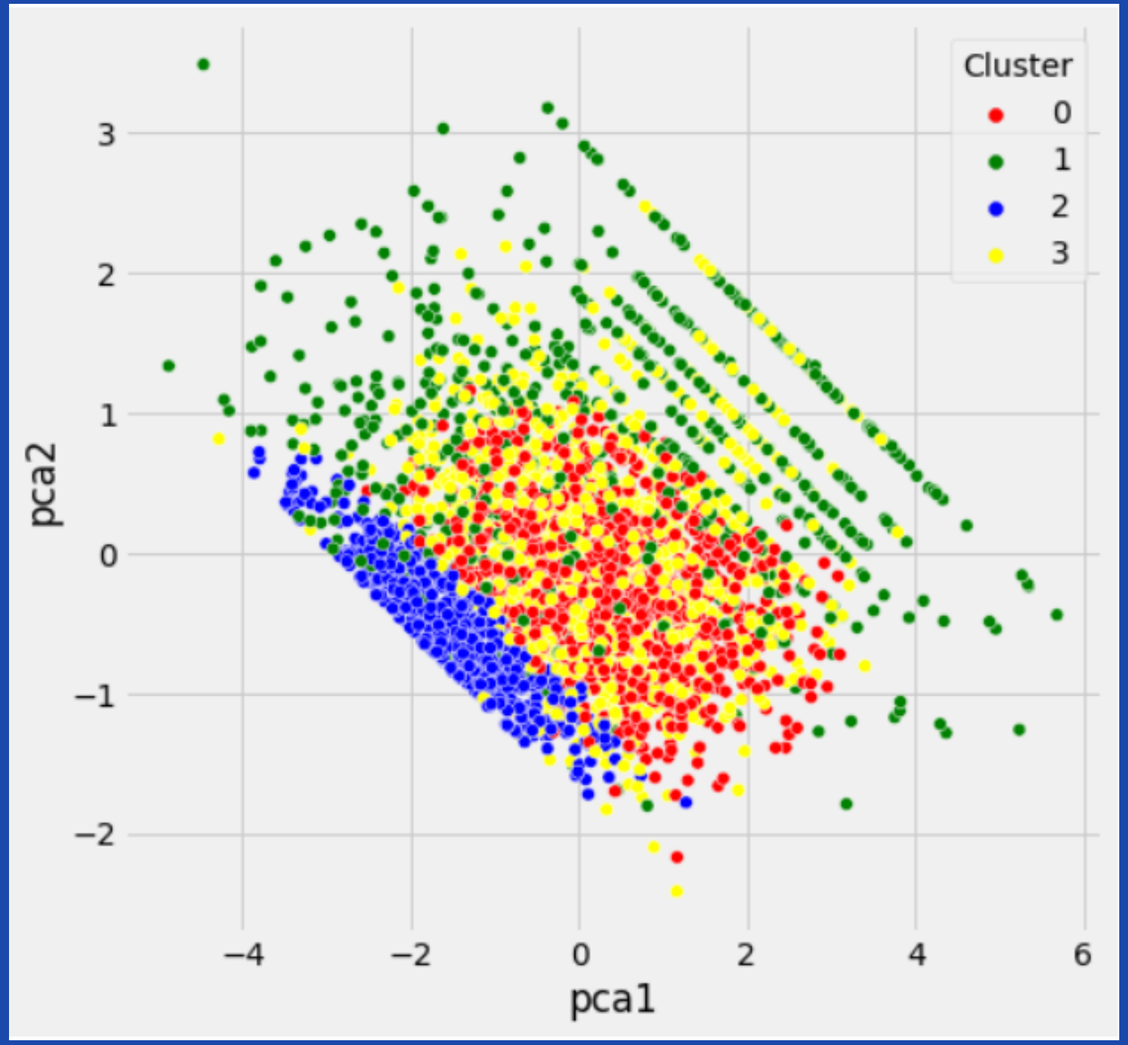
Cluster 2 --> Green
--> It includes approx 80 perc. most loyal customers along with less loyal ones i.e. gold and very less number of silver type.

Cluster 3 --> Yellow
--> It includes most un-loyal customers(at churning out phase) along with less loyal ones i.e. gold and silver types.



Clusters	Platinum	Gold	Silver	Bronze
Cluster 0	0	46	90	149
Cluster 1	364	165	78	3
Cluster 2	774	448	1	0
Cluster 3	2	522	553	726

PCA applied on Gaussian Mixture Model Selection



Clustering Performance Evaluation

Silhouette Coefficient : If the ground truth labels are not known, evaluation must be performed using the model itself. The Silhouette Coefficient is an example of such an evaluation, where a higher Silhouette Coefficient score relates to a model with better defined clusters.

Hierarchical clustering is having the highest Silhouette Coefficient

-> HIERARCHICAL CLUSTERING

```
# K-Means Clustering
metrics.silhouette_score(Scaled_Data, labels_kmeans, metric='euclidean')
```

```
0.2965916610480074
```



```
# DBSCAN Clustering
metrics.silhouette_score(scale[['Recency','Frequency']], labels_dbscan, metric='euclidean')
```

```
-0.04439954121784424
```

```
# Hierarchical Clustering
metrics.silhouette_score(Scaled_Data, labels_hierarchial, metric='euclidean')
```

```
0.36165618921298065
```

```
# Gaussian Mixure Model
metrics.silhouette_score(Scaled_Data, cluster, metric='euclidean')
```

```
0.187811730026961
```

Clustering Performance Evaluation

Calinski-Harabasz Index : If the ground truth labels are not known, the Calinski-Harabasz index , also known as the Variance Ratio Criterion - can be used to evaluate the model, where a higher Calinski-Harabasz score relates to a model with better defined clusters.

```
# K-Means Clustering
metrics.calinski_harabasz_score(Scaled_Data, labels_kmeans)
```

```
2620.837834676728
```

```
# DBSCAN Clustering
metrics.calinski_harabasz_score(scale[['Recency','Frequency']], labels_dbscan)
```

```
70.97408705315446
```

Hierarchical clustering is having the highest Calinski-Harabasz Index

```
# Hierarchical Clustering
metrics.calinski_harabasz_score(Scaled_Data, labels_hierarchial)
```

```
2988.7123856283292
```

-> HIERARCHICAL CLUSTERING

```
# Gaussian Mixure Model
metrics.calinski_harabasz_score(Scaled_Data, cluster)
```

```
995.6789111739154
```

Clustering Performance Evaluation

Davies-Bouldin Index : If the ground truth labels are not known, the Davies-Bouldin index can be used to evaluate the model, where a lower Davies-Bouldin index relates to a model with better separation between the clusters.

Hierarchical clustering is having the lowest Davies-Bouldin Index

-> HIERARCHICAL CLUSTERING

```
# K-Means Clustering
metrics.davies_bouldin_score(Scaled_Data, labels_kmeans)
```

```
1.1146492449047696
```

```
# DBSCAN Clustering
metrics.davies_bouldin_score(scale[['Recency','Frequency']], labels_dbscan)
```

```
1.4509974419162226
```

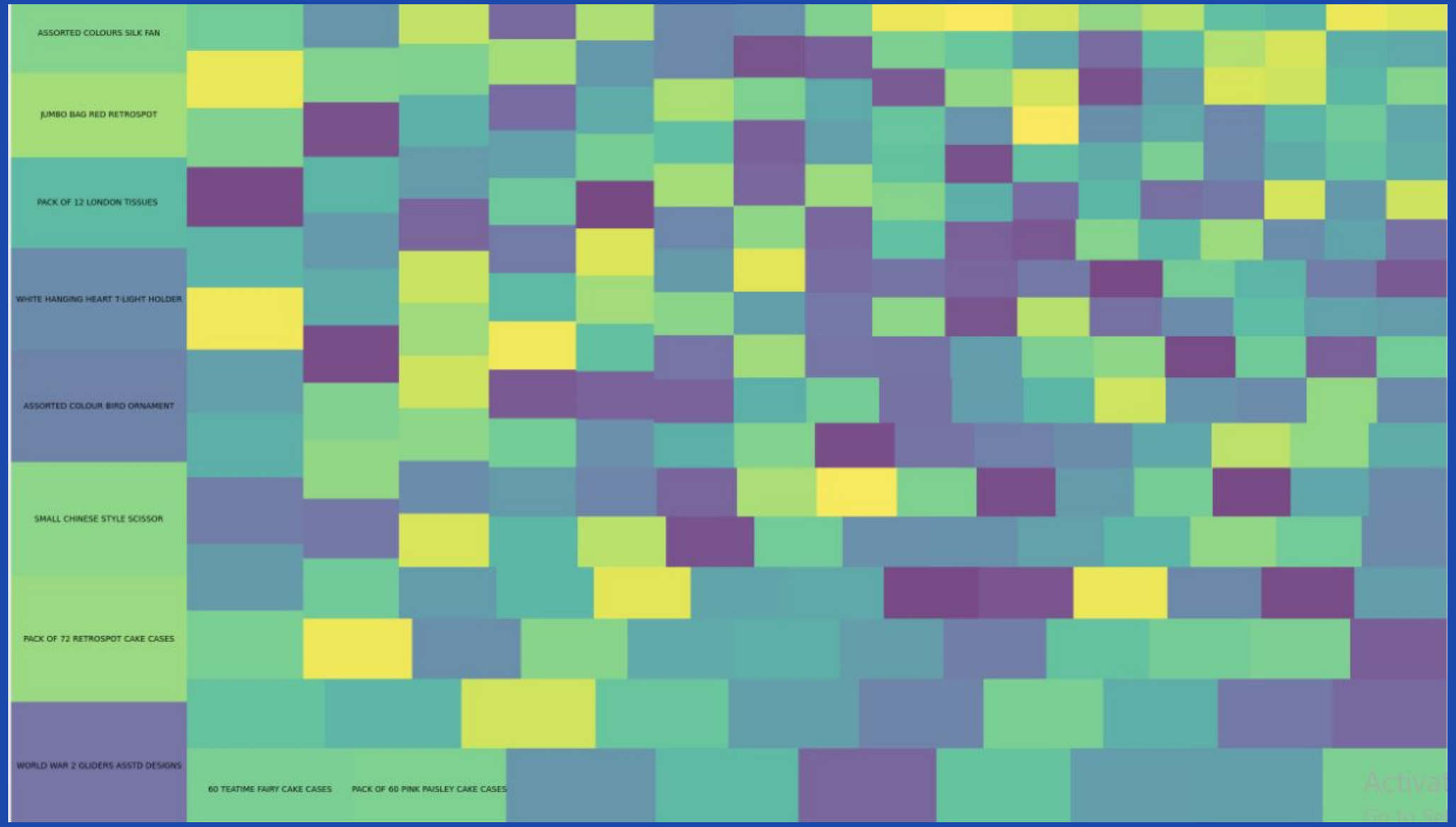
```
# Hierarchical Clustering
metrics.davies_bouldin_score(Scaled_Data, labels_hierarchial)
```

```
1.0084176232379798
```

```
# Gaussian Mixure Model
metrics.davies_bouldin_score(Scaled_Data, cluster)
```

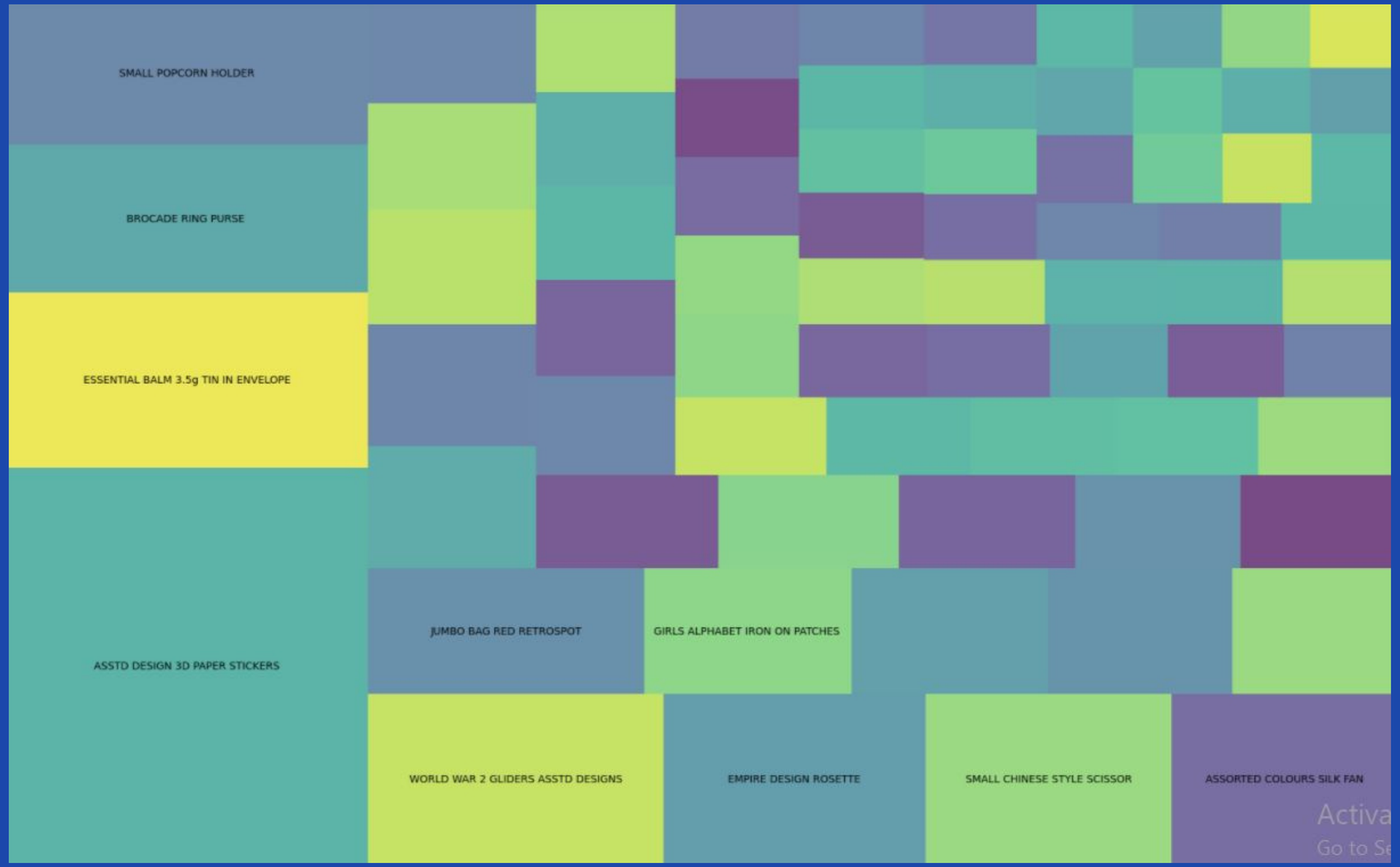
```
1.6178891379171583
```

Most Frequent Products in each Cluster (K-Means-cluster-1) TreeMap Implementation



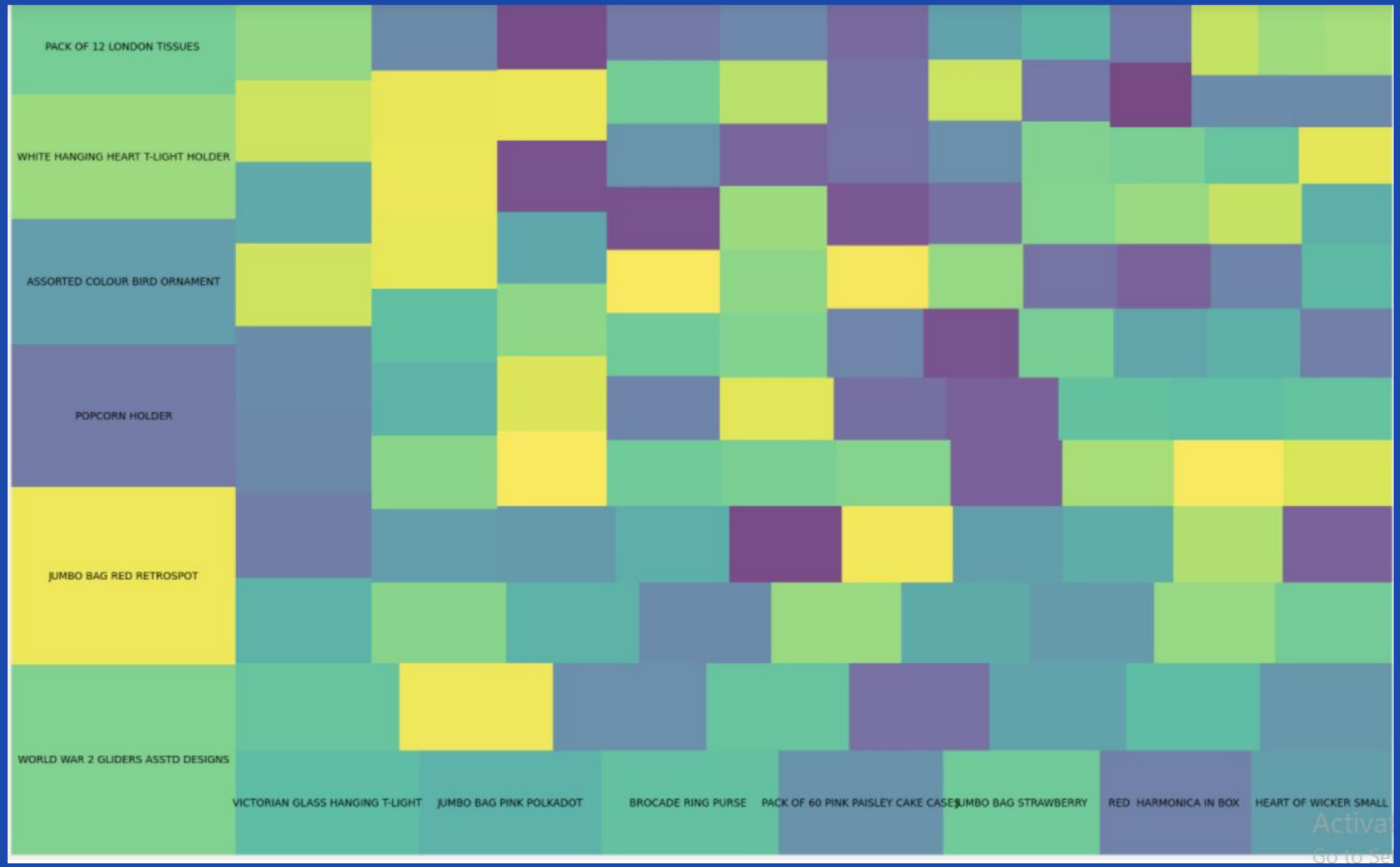
Most Frequent Products in each Cluster

(K-Means-cluster-2)



Most Frequent Products in each Cluster

(K-Means-cluster-3)



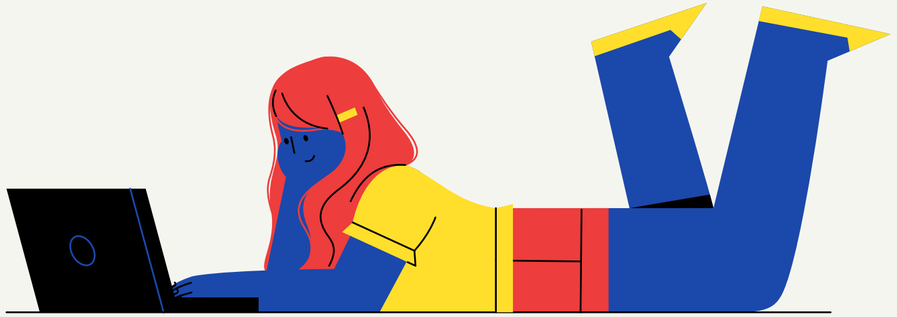
From the above Treemap implementation on each cluster of K-means clustering, we can observe that:

1. In cluster 1, Most frequent items purchased by customers are World War 2 Gliders, Teamtime fairy cake cases,pack of 60 paisley fairy cake cases,pack of 72 retrospot cake cases,small chinese size scissor,assorted color bird ornament,T-light holder etc.
2. In cluster 2 , Most Frequent Items were 3D paper stickers,world war 2 gliders,empire design rosette,small chinese style scissor,color silk fan,essential balm 3.5g in envelope,brocade ring purse and small popcorn holder.
3. In cluster 3, Most Frequent Items were world war 3 gliders,victorian glass hanging T-light,jumbo bag red retrospot,popcorn holder,jumbo bag pink polkadot,brocade ring purse,assorted color bird ornament,pack of 12 london tissues etc.

The most common item in all three clusters is world war 2 gliders asstd. designs

Research Paper

https://www.overleaf.com/project/607d448fe8d9d71278cfd805



THANK YOU!