TABLE OF CONTENTS

1.ABSTRACT

Customer relationship management is a marketing concept, which consists customer identification, attraction, retention and development. We can say that CRM is a strategy of acquiring, retaining customers to get high value for the business and the customer. CRM strategy provides integration of technology, processes and all business works related to the customers. This project investigates how to integrate algorithms and Analysis of Recency, Frequency, and Monetary value.This analysis can be done in online trading to provide strategies based on customer purchasing behavior. To maintain and promote the business, many aspects are important to be considered to make sure that benefits increase. Customer segmentation helps businesses to target their services and prioritizing products on the basis of its gain. Success of an organization depends on attracting and keeping loyal customers. In this project, we applied various Clustering Algorithms on Customers data to segment the customers into clusters in which each cluster has definite characteristics. By comparing the above algorithms, we can find key customer clusters based on Recency, Frequency, Monetary values.

2.INTRODUCTION

The RFM is used in the analysis of customer values. It consists of three values. The first is "Recency" which tells how recent the customer has made a transaction or purchase. Second is "Frequency" which tells how many times did the customers make a transaction or purchase. The Third is "Monetary" which is the sum of all amounts that the customer has spent buying. Using these three values, clusters are to be made by considering the scaled values of RFM. Each cluster has its definite characteristics. By understanding the characteristics of each clusters, customer groups can be identified.

2.1 MACHINE LEARNING

Machine learning makes the system automatically learn and also to improve through experience without the need for being separately programmed.

Machine learning emphasizes on generation of computer programs that can use data to learn. Machine Learning is related to how to construct programs which automatically gets improved through experience.

2.1.1 Supervised Machine Learning

In Supervised Machine Learning, the data is represented as labelled and the algorithm gets learned from the training data that is labelled.

Classification and Regression are the examples. Supervised Machine learning uses training data to teach the models to give the required output.

The training dataset does includes input values and correct output values of the attributes, that allows the model to get learned over the time.

2.1.2 Unsupervised Machine Learning

Contrary to Supervised Machine Learning, Unsupervised Machine Learning uses the data that is not labelled.

From that unlabelled data, it finds patterns which help to solve clustering problems.

Examples: Hierarchical Clustering, K-Means Clsutering.

2.2 CLUSTERING

Clustering is an unsupervised machine learning algorithm. It's used for finding natural clusters in the dataset. Clustering algorithms analyse the input data and discover the natural clusters in the data.

2.3 K-MEANS CLUSTERING

Unlike supervised learning algorithms, K-means clustering is an unregulated machine learning algorithm. K-Means is used when we have non-labeled data (Data with no categories or groups). Our customer segregation data is like this problem.

The algorithm finds Clusters in the data, in which the number of clusters is represented by the value of K. The algorithm acts repetitively to give each data point to one of K clusters, according to the features. This makes K-Means suitable for the Customer Categorization problems.

Given a set of data points are grouped as per feature similarity.

At the end, we are going to get various clusters along with cluster ids to which the customer belongs to.

2.4 AGGLOMERATIVE CLUSTERING

Agglomerative clustering is the most popular clustering algorithm that is used to classify points according to their similarity. The algorithm begins by considering each item as a singleton group. Next, the pairs of groups are grouped into sequences until all the clusters have been grouped together into a large collection which consists of all the items. The output is a tree representation of elements, known as a dendrogram.

Agglomerative clustering is a “bottom-up” approach. Each object is initially considered as a single element (leaf). In each step of the algorithm, two very similar groups are grouped together into a large group. This process is repeated until all the points become part of one large group (root).

To determine which items / collections should be merged or separated, we need the methods to measure similarities between elements.

There are ways to calculate similarity, like Euclidean distance and Manhattan distance.

The linkage function considers distances and then combines pairs of elements into clusters based on the similarity. These newly formed groups are linked to one another to form larger collections. This steps are repeated until all the elements in the dataset are connected together in the tree.

2.5 DBSCAN

DBSCAN is a widely used clustering algorithm which is used for data mining and machine learning. According to the set of points, DBSCAN collects adjacent points according to the distance measurement like Euclidean distance and a minimum points paarmeter. It marks the exterior of points in densely populated areas.

The DBSCAN Clustering algorithm requires two parameters:

Epsilon: Represents how close the points should be in order to be considered as part of a cluster. This means that if the Euclidean distance between two points is less than or equal to Epsilon, these points are considered to be neighbors.

minPoints: The minimum number of data points in order to form a cluster.

2.6 MEANSHIFT CLUSTERING

Mean shift is an unsupervised Machine Learning clustering algorithm which is known to be used for clustering. It is mostly used in image segmentation because it does not have any parameters and does not require any shape of the clusters to be pre-defined in the feature space.

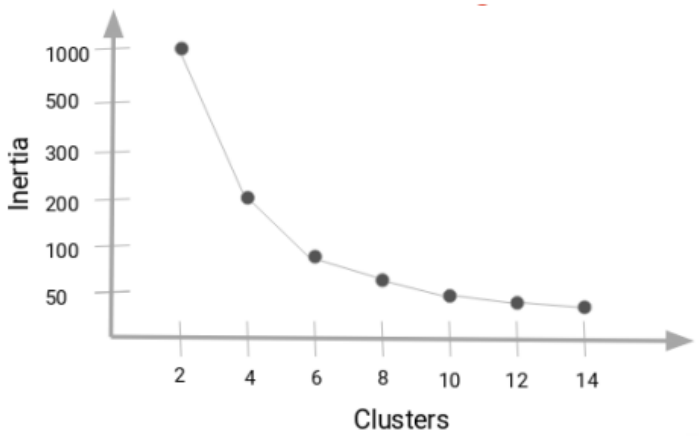
"Mean Shift" means "Shifting to the Mean" in an iterative manner. In the algorithm, every data point shifts to the "regional mean" interatively and the location of the final position of each point represents the cluster to which it belongs to.

In mean shift algorithm, each point tries to find its cluster by moving towards the weighted mean of its local area in each iteration. The final position of each point will be the centroid of the cluster that the point belongs to. Then, all the data points with the final position point can be labeled with the same cluster.

2.7 EVALUATION METRICS

2.7.1 Elbow Curve

Elbow Curve is used in Hyperparameter Tuning to find the optimal number of clusters for the Model.



Example Image for Elbow curve

2.7.2 INERTIA

In this, we calculate IntraCluster Distances.

Calculate the Sum of all the Distances of all the points within a cluster from the centroid.

Calculate this sum for all the clusters and the total inertia value is the sum of all these intracluster distances.

Inertia is equal to the sum of intracluster distances.

Lesser the inertia value, the better our clusters are.

2.7.3 Silhouette Coefficient:

The Silhouette Score is calculated using two values

(a) The Average of all the intra-cluster distances

(b) The Average of all the nearest-cluster distances for datapoints.

The Silhouette Score -> (b - a) / maximum(a, b).

b -> The distance between a data point and the cluster that is nearest to which data point that does not belong to.

One is the desirable and best value and the worst value is -1. Values near 0 indicate that the clusters are overlapping. Negative values normally indicate that a observation has been assigned to a wrong cluster of data points.

2.8 Dataset

Name : onlineretail.csv

Source: UCI Machine Learning Repository

Attributes

- “InvoiceNo”
- “StockCode”
- “Description”
- “Quantity”
- “InvoiceDate”
- “UnitPrice”
- “CustomerID”
- “Country”

3. LITERATURE REVIEW

This study has built various clustering models based on the customer data according to their Internet Banking usage for the categorization of customers in XYZ bank by using K-Means clustering algorithm and K-Medoids algorithm. The total performance of the clustering was measured and compared with other algorithms. K-Means Algorithm was efficient than the K-Medoids algorithm according to the average intra cluster distances.[2]

"The suggested method has concluded that for the task of clustering, when we merge the RFM modelling with the K-means method, we can see a very high improvement in the task of classifying accuracy for reaching to a marvellous CRM."[1]

"In the past, banks, retailers, insurance companies had close ties with their customers and knew what customers want; so they tried to grant their needs and wishes by offering special services to them. Later, with the arrival of mass production and marketing and increased number of consumer customers, the importance of building relationships with customers was reduced and the variety of products and their prices also declined. There are a lot of evidences indicating that the customer relationships were taken into consideration since the late 19th century. Today, through the effective use of information technology and communication, organizations can provide their customers with diverse products with lower prices and special services simultaneously. One of the most effective tools to study customers' behaviour is using clustering techniques. In this chapter, the use of segmentation to segment customers and its importance are described, then questions and related objectives are developed and finally the main research variables are defined."[3]

Machine learning methods, LORM, LIRM and NEM, are implemented for segmenting customers dataset that belongs to a company. The dataset has two dimensions that are number of payments and total amount of payments by each one of the customers. This research has proposed to solve a problem of customer segmentation of an organization by using payment

information of the customers. The outputs of this study can be implemented in the new age CRM software.[4]

4. PROBLEM IDENTIFICATION AND OBJECTIVES

Customer Relationship Management marketing strategy considers the integration of technology in the processes of businesses, related to the customer.

There is a need for the businesses and organizations to segments customers based on the characteristics of their customer base. As customer Loyalty and Retention are the important objectives of any business, so the main aim of this project is to provide an effective and efficient implementation the organization's objective to segment the customers according to the RFM Analysis.

5. SYSTEM METHODOLOGY

5.1 EXPLORING THE DATASET

Data exploration is the initial step in the methodology in which we use data visualization techniques to describe characterizations of the dataset.

Data exploration techniques explore and identify relationships between different attributes.

This analysis is used to identify patterns that gives access to insight into the raw data.

5.2 DATA CLEANING

Data cleaning is a process of removing or changing or fixing false, corrupted and incomplete data in the dataset.

While merging multiple data sources, there are many ways for data to be duplicated or mislabelled.

If the data is false, outcomes and algorithms are not reliable.

5.3 DATA VISUALIZATION

Data is represented in the graphical format using maps, charts, graphs.

This provides an way to see and understand outliers, and patterns in dataset.

5.4 DATA PREPROCESSING

Pre-processing of data is used to check the quality of data. The quality can be known by the following

- **Accuracy**
- **Completeness**
- **Consistency**
- **Timeliness**
- **Believability**
- **Interpretability**

5.5 EXTRACTING THE VALUES OF R,F,M

Recency:

RECENCY value is calculated in number of days.

Take maximum date as [maximum date in the dataset + 1]

Calculate the difference between the maximum date and the date of recent transaction of the customer.

The lower this value, the best is the customer.

Frequency:

FREQUENCY refers to the number of times the customer has made a transaction/

Calculation follows as the number of invoices of each customer.

The higher the value of frequency, the best is the customer

Monetary Value:

MONETARY Value is the total amount spent by the customer.

The product of UNIT PRICE and QUANTITY gives the value of Monetary.

6. OVERVIEW OF TECHNOLOGIES

6.1 PYTHON

Python is an advanced programming language that supports the development of many applications. Python Language comes with many libraries and frameworks that make typing easier. This also saves valuable time.

The popular NumPy libraries, used for scientific statistics; SciPy to get the most advanced statistics; and scikit, for data mining and data analysis.

Python code is short and readable even for new developers, which benefits machine and in-depth learning projects. Because of its simple syntax, the development of Python applications is faster compared to most programming languages.

6.2 NUMPY

NumPy -> Numerical Python.

NumPy is a Python library used to work with arrays.

It has methods for working on line algebra line and matrices.

In Python we have lists that serve the purpose of the array, but are slow to process.

NumPy aims to provide something up to 50x faster than a standard Python list.

NumPy's list item is called *ndarray*, it provides many support functions which works with ndarray much easier.

Arrays are widely used in data analysis, where speed and memory are atmost important.

6.3 MATPLOTLIB

Matplotlib is used to plot graphs in python that works as a visualization tool. Matplotlib is an open source library and we can use it for free.

Matplotlib is written in python, C, JavaScript.

6.4 PANDAS

Pandas is a library built for Python for Data Analysis and Manipulation. It consists of various data structures and operations to analyse and manipulate Data.

In this project, we use Pandas for exploring Dataset and manipulating the data.

7. IMPLEMENTATION

7.1 EXTRACTING R F M Values

R Values:

In [92]:
```
recency  = data[['CustomerID','InvoiceDate']]


maximum = max(recency.InvoiceDate)

maximum = maximum + pd.DateOffset(days=1)
recency['diff'] = maximum - recency.InvoiceDate
recency.head()
```

```
<ipython-input-92-f80f3eafba61>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  recency['diff'] = maximum - recency.InvoiceDate
```

Out[92]:

	CustomerID	InvoiceDate	diff
0	17850.0	2010-12-01 08:26:00	374 days 04:24:00
1	17850.0	2010-12-01 08:26:00	374 days 04:24:00
2	17850.0	2010-12-01 08:26:00	374 days 04:24:00
3	17850.0	2010-12-01 08:26:00	374 days 04:24:00
4	17850.0	2010-12-01 08:26:00	374 days 04:24:00

F Values:

In [246]:
```
frequency = data[['CustomerID', 'InvoiceNo']]
```

In [247]:
```
k = frequency.groupby("CustomerID").InvoiceNo.count()
k = pd.DataFrame(k)
k = k.reset_index()
k.columns = ["CustomerID", "Frequency"]
k.head()
```

Out[247]:

	CustomerID	Frequency
0	12346.0	2
1	12347.0	182
2	12348.0	31
3	12349.0	73
4	12350.0	17

M Values:

In [245]:
```
data = pd.concat(objs = [data, monetary], axis = 1, ignore_index = False)

# Finding total amount spent per customer
monetary = data.groupby("CustomerID").Monetary.sum()
monetary = monetary.reset_index()
monetary.head()
```

Out[245]:

	CustomerID	Monetary
0	12346.0	0.00
1	12347.0	4310.00
2	12348.0	1797.24
3	12349.0	1757.55
4	12350.0	334.40

Combining RFM

In [254]:
```
RFM = k.merge(monetary, on = "CustomerID")
RFM = RFM.merge(df, on = "CustomerID")
RFM.head()
```

Out[254]:

	CustomerID	Frequency	Monetary	Recency
0	12346.0	2	0.00	326 days 02:33:00
1	12347.0	182	4310.00	2 days 20:58:00
2	12348.0	31	1797.24	75 days 23:37:00
3	12349.0	73	1757.55	19 days 02:59:00
4	12350.0	17	334.40	310 days 20:49:00

7.2 SCALING THE DATA

In [26]:
```
RFM_norm = RFM.drop(["CustomerID"], axis=1)
RFM_norm.Recency = RFM_norm.Recency.dt.days

from sklearn.preprocessing import StandardScaler
standard_scaler = StandardScaler()
RFM_norm = standard_scaler.fit_transform(RFM_norm)
```

In [27]:
```
RFM_norm = pd.DataFrame(RFM_norm)
RFM_norm.columns = ['Frequency','Amount','Recency']
RFM_norm.head()
```

Out[27]:

	Frequency	Amount	Recency
0	-1.070949	-1.041614	2.136422
1	-0.375498	1.385298	-0.282050
2	0.631707	1.331702	-0.821629
3	-0.711233	-0.590055	1.982257
4	1.159290	1.045238	-0.657828

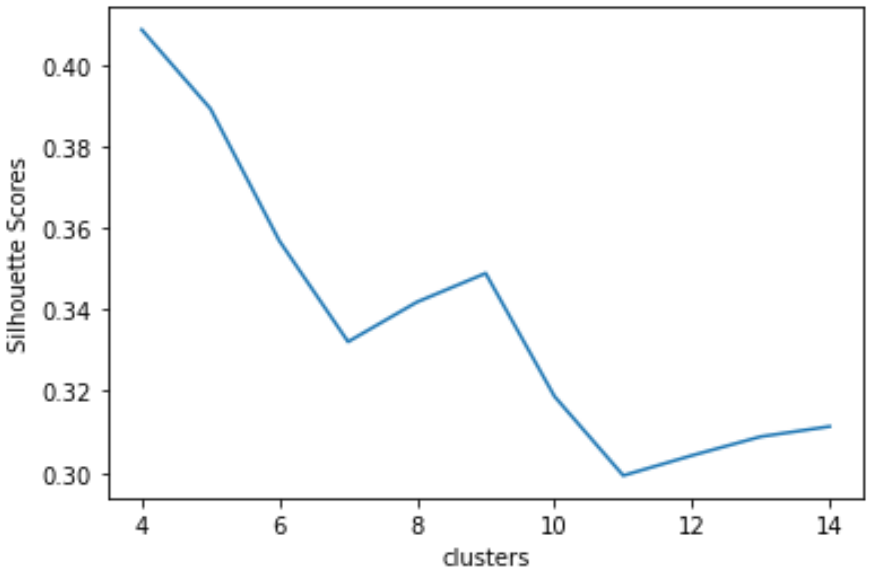
7.3 KMEANS CLUSTERING

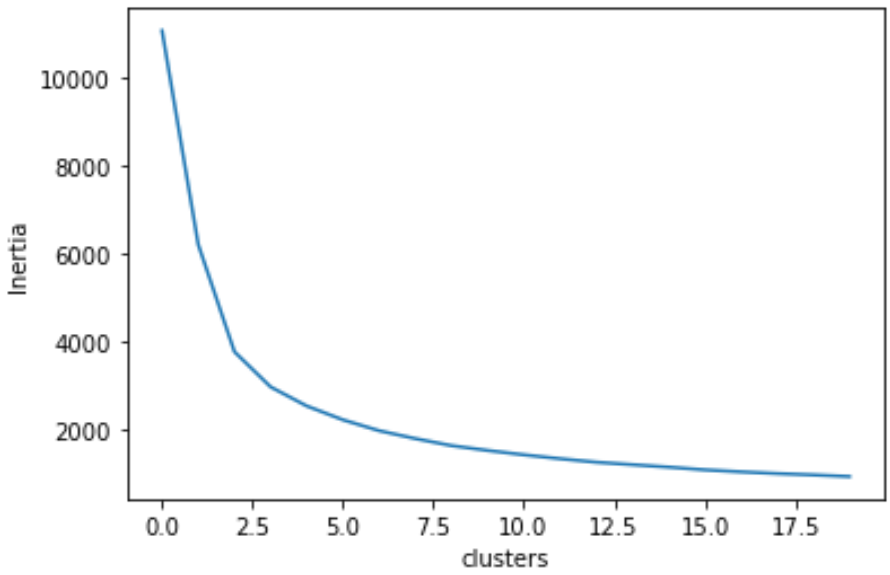
7.3.1 ALGORITHM

- Randomly initialize K observations as initial centroids.
- Assign all observations into K groups based on their distance from K clusters meaning assign observation to the nearest cluster.
- Take the mean of all observations present in clusters and then make that mean as the centroid of the respective cluster.
- Perform multiple iterations of 2 and 3 until our clusters are not determined.

7.3.2 FINDING K VALUE

The value of K represents the number of clusters to be formed.





THE OPTIMAL VALUE OF K IS 4

Silhouette Coefficient:

The Silhouette Score is calculated using two values

(a) The Average of all the intra-cluster distances

(b) The Average of all the nearest-cluster distances for datapoints.

The Silhouette Score -> (b - a) / maximum(a, b).

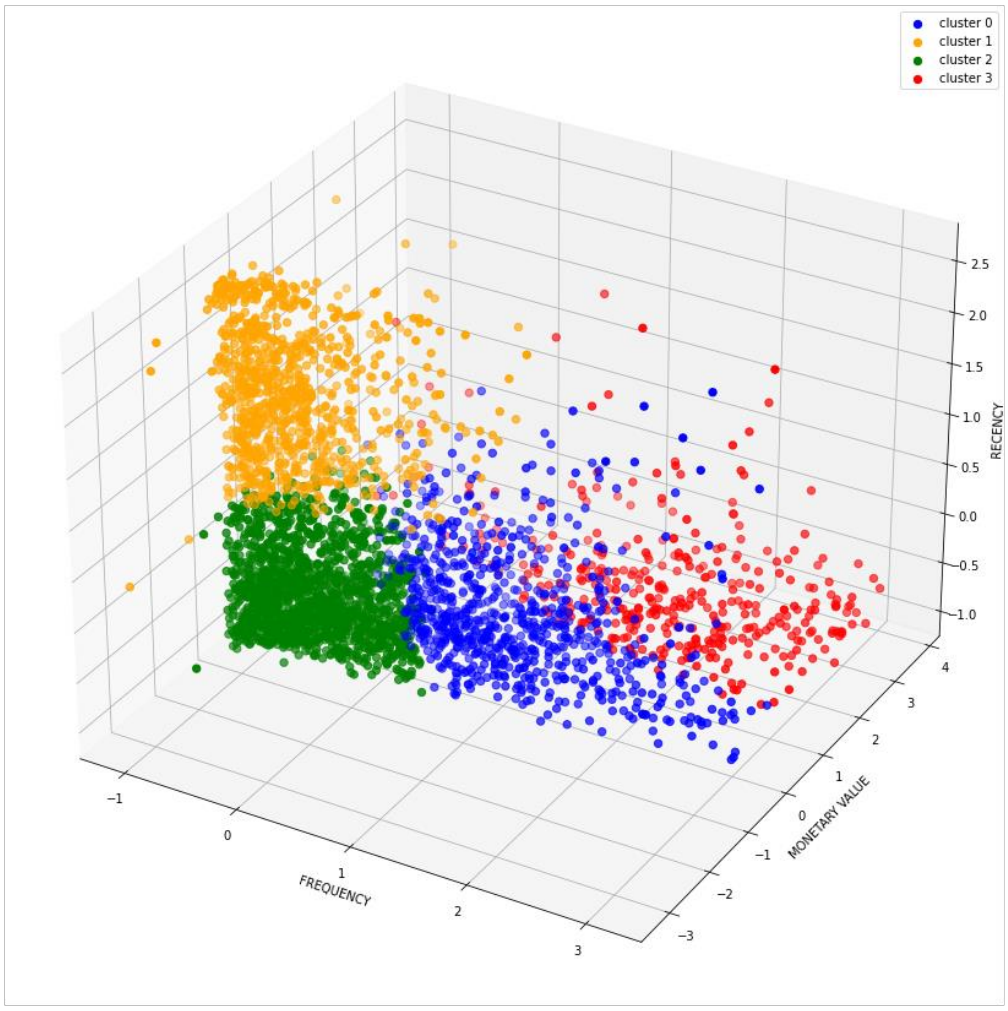
b -> The distance between a data point and the cluster that is nearest to which data point that does not belong to.

- The best value -> 1
- The Worst value -> -1
- Values approaching 0 indicate that the clusters are overlapping.
- Negative values -> The point is given to a wrong cluster.

7.3.3 IMPLEMENTING K MEANS CLUSTERING

```
In [31]: kmeans_model = KMeans(n_clusters=4, init='k-means++', random_state=42)
         y_kmeans = kmeans_model.fit_predict(RFM_norm)
         X = np.array(RFM_norm)
```

```
In [32]: fig = plt.figure(figsize = (15,15))
         ax = fig.add_subplot(111, projection='3d')
         ax.scatter(X[y_kmeans == 0,0],X[y_kmeans == 0,1],X[y_kmeans == 0,2], s = 40 , color = 'blue', label = "cluster 0")
         ax.scatter(X[y_kmeans == 1,0],X[y_kmeans == 1,1],X[y_kmeans == 1,2], s = 40 , color = 'orange', label = "cluster 1")
         ax.scatter(X[y_kmeans == 2,0],X[y_kmeans == 2,1],X[y_kmeans == 2,2], s = 40 , color = 'green', label = "cluster 2")
         ax.scatter(X[y_kmeans == 3,0],X[y_kmeans == 3,1],X[y_kmeans == 3,2], s = 40 , color = 'red', label = "cluster 3")

         ax.set_xlabel('FREQUENCY')
         ax.set_ylabel('MONETARY VALUE')
         ax.set_zlabel('RECENCY')
         ax.legend()
         plt.show()
```



NUMBER OF CLUSTERS = 4

7.3.4 ASSIGNING CLUSTER IDs TO CUSTOMERS

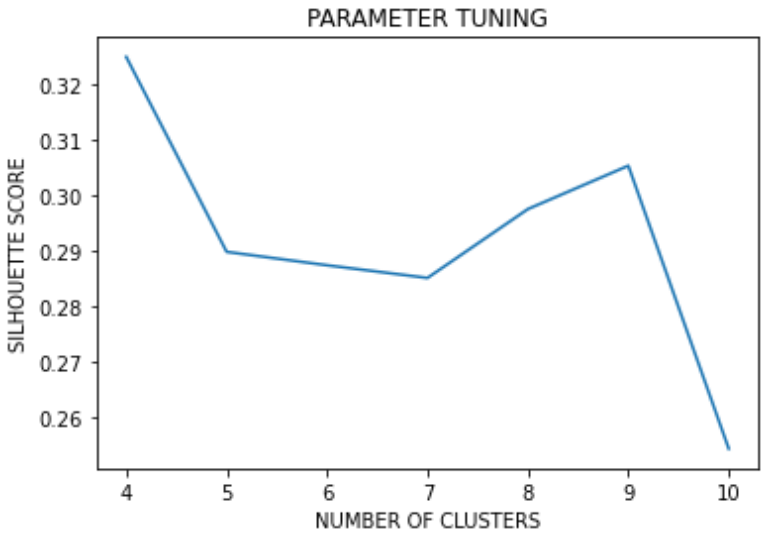
In [97]:
```
RFM_km = pd.concat([RFM, pd.Series(y_kmeans)], axis=1)
RFM_km.columns = ['CustomerID', 'Frequency', 'Monetary', 'Recency', 'ClusterID']

RFM_km.Recency = RFM_km.Recency.dt.days
km_clusters_monetary = pd.DataFrame(RFM_km.groupby(["ClusterID"]).Monetary.mean())
km_clusters_frequency = pd.DataFrame(RFM_km.groupby(["ClusterID"]).Frequency.mean())
km_clusters_recency = pd.DataFrame(RFM_km.groupby(["ClusterID"]).Recency.mean())
RFM_km.head()
```

Out[97]:

	CustomerID	Frequency	Monetary	Recency	ClusterID
0	12346.0	2	0.00	326	1
1	12348.0	31	1797.24	75	0
2	12349.0	73	1757.55	19	0
3	12350.0	17	334.40	310	1
4	12352.0	95	1545.41	36	0

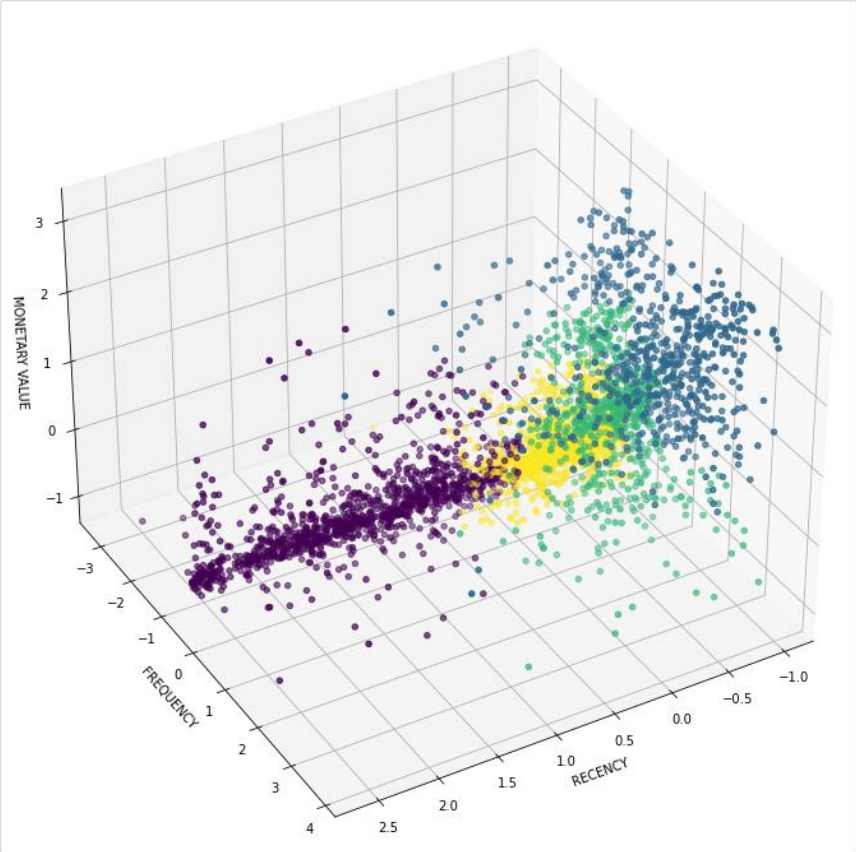
7.4 AGGLOMERATIVE CLUSTERING

7.4.1 Finding Number of Clusters:



7.4.2 Implementing Agglomerative Clustering:

```
In [39]: model = AgglomerativeClustering(n_clusters=4, affinity='euclidean', linkage='ward')
         agg=model.fit_predict(RFM_norm)
         agg

Out[39]: array([0, 2, 1, ..., 0, 3, 1], dtype=int64)
```



7.5 DBSCAN

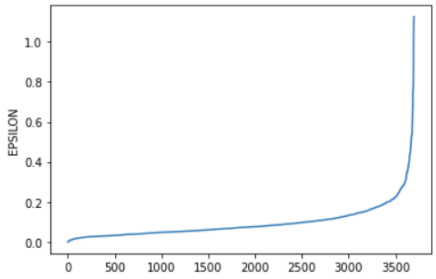
7.5.1 FINDING EPSILON

In [106]:
```
from sklearn.neighbors import NearestNeighbors
```

In [107]:
```
neigh = NearestNeighbors(n_neighbors=2)
nbrs = neigh.fit(RFM_norm)
distances, indices = nbrs.kneighbors(RFM_norm)
```

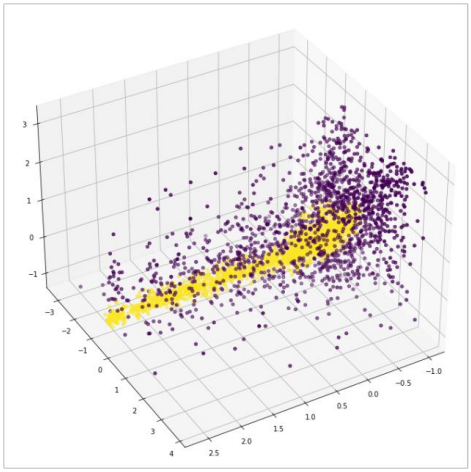
In [108]:
```
distances = np.sort(distances, axis=0)
distances = distances[:,1]
plt.ylabel("EPSILON")
plt.plot(distances)
distances
```

Out[108]:
```
array([1.35035492e-04, 1.35035492e-04, 1.01276619e-03, ...,
       1.10735719e+00, 1.12508508e+00, 1.12508508e+00])
```



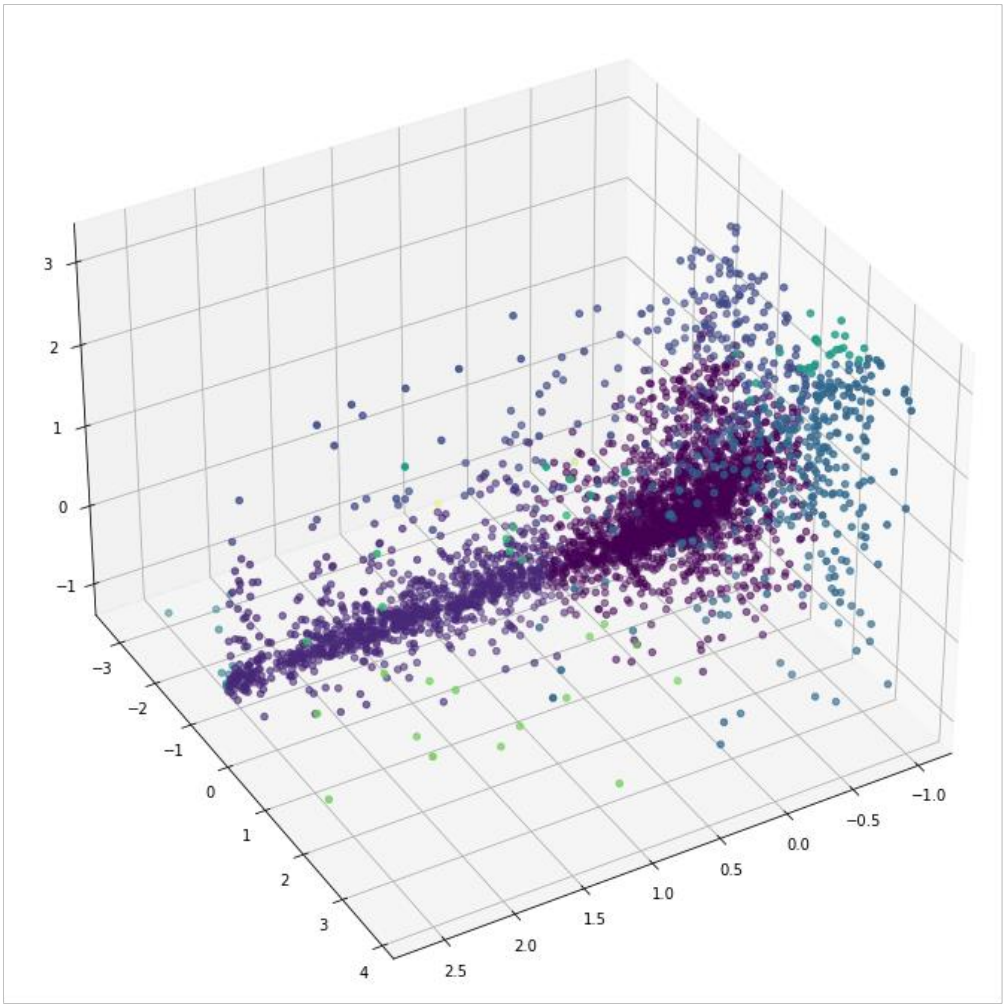
7.5.2 IMPLEMENTING DBSCAN

In [109]:
```
dbsc = DBSCAN(eps = 0.21, min_samples = 20).fit(RFM_norm)
dbs=dbsc.fit_predict(RFM_norm)
```



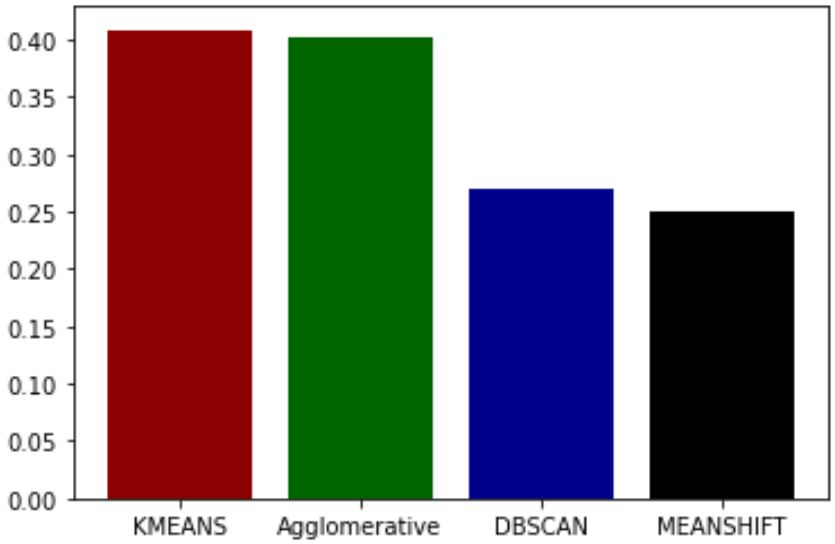
7.6 MEANSHIFT

```
In [120]: from sklearn.cluster import MeanShift

          ms = MeanShift(bandwidth=1)
          msh = ms.fit_predict(RFM_norm)
          msh

Out[120]: array([1, 0, 0, ..., 1, 0, 0], dtype=int64)
```



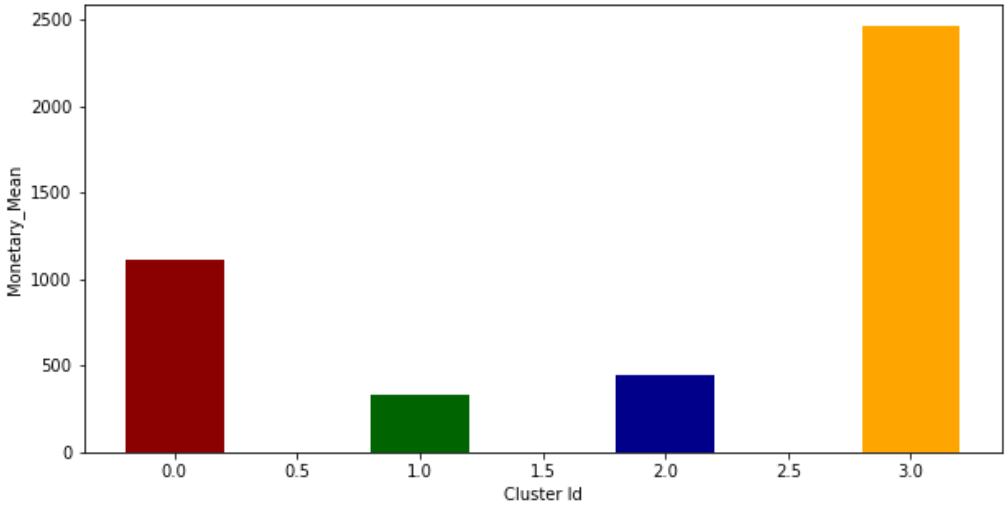
7.7 Comparing Models using Silhouette Scores

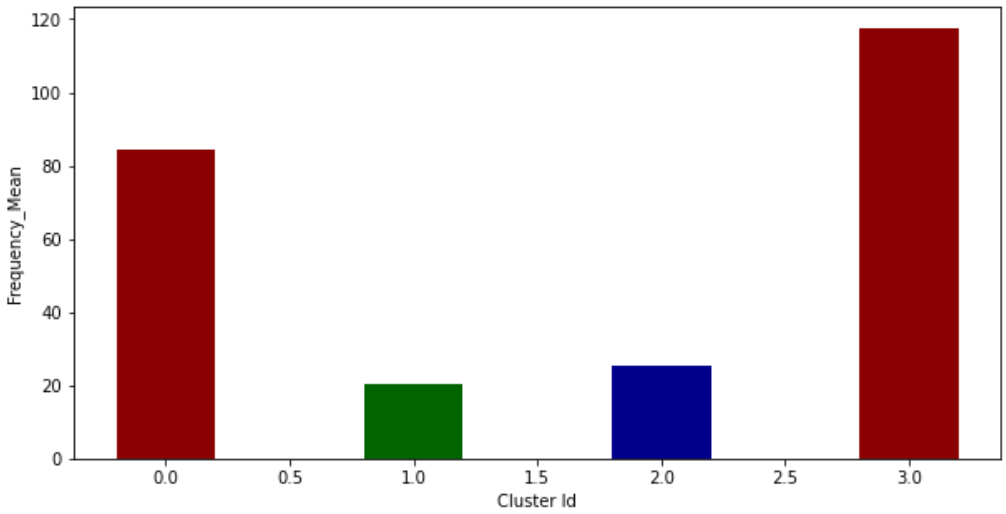
- K-MEANS → 0.40851721359807264
- AGGLOMERATIVE CLUSTERING → 0.3249143321884565
- DBSCAN → 0.2692400279336455
- MeanShift → 0.2505909231704478

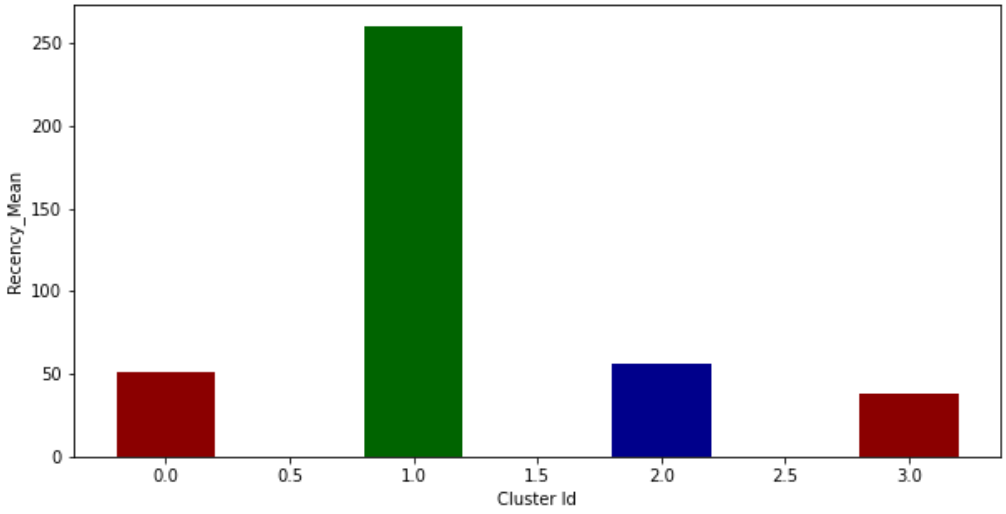


KMEANS GOT HIGHEST SILHOUETTE SCORE

7.8 FINAL CATEGORIES







CLUSTER 0 : Moderate Monetary value, High Frequency value, Low Recency value - GOOD VALUE LOYAL CUSTOMERS

CLUSTER 1 : Low Monetary value, Low Frequency value, High Recency value - LOST UNSATISFIED CUSTOMERS

CLUSTER 2 : Low Monetary value,Low Frequency value, Low Recency value - UNSATISFIED CUSTOMERS

CLUSTER 3 : High Monetary value, High Frequency Value, Low Recency Value - BEST VALUE CUSTOMERS

8. RESULTS

After implementing the clustering algorithms KMeans, Agglomerative, DBSCAN and Meanshift, we found that the KMeans algorithm shows good quality clusters over others. So, segments were done based on the results given by KMeans clustering.

Four segments were formed each of which has definite characteristics.

9. CONCLUSION

This Project built various clustering models on Retail Customer dataset based on Recency and spending behaviours of the customers for customer segmentation.

The performances of all the clustering models was measured and are compared. K-Means model outperformed Agglomerative Clustering, DBSCAN and Meanshift Clustering based on Silhouette Scores.

The optimal number of clusters for KMeans is 4, according to the highest value of Silhouette Score.

10. REFERENCES

[1] I. Maryani, D. Riana, R. D. Astuti, A. Ishaq, Sutrisno and E. A. Pratama, "Customer Segmentation based on RFM model and Clustering Techniques With K-Means Algorithm," 2018 Third International Conference on Informatics and Computing (ICIC), 2018, pp. 1-6, doi: 10.1109/IAC.2018.8780570.

[2] M. Aryuni, E. Didik Madyatmadja and E. Miranda, "Customer Segmentation in XYZ Bank Using K-Means and K-Medoids Clustering," 2018 International Conference on Information Management and Technology (ICIMTech), 2018, pp. 412-416, doi:10.1109/ICIMTech.2018.8528086.

[3] Karim Hamdi and Ali Zamiri, 2016. Identifying and Segmenting Customers of Pasargad Insurance Company Through RFM Model (RFM). International Business Management, 10: 4209-4214.

[4] Ş. Ozan, "A Case Study on Customer Segmentation by using Machine Learning Methods," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1-6, doi: 10.1109/IDAP.2018.8620892.

[5] Nimbalkar, Divya & Shah, Paulami. (2013). Data mining using RFM Analysis. 10.13140/RG.2.2.24229.04328.

[6] https://towardsdatascience.com/how-to-build-a-machine-learning-model-439ab8fb3fb1

[7] https://www.tableau.com/learn/articles/data-visualization