UnSupervised Learning Competition Report

❖ Course Details

- Course Name: Data Analytics
- Course ID: CISC-839
- Course Instructor: Hazem Abbas

❖ Group Members

- Karthik Ranga Swamy: 20188884
- Abhishek Awasthi: 20143052
- Dhanush Reddy Nandyala: 20134832

❖ Project Topic

- Customer Segmentation for Online Retail

❖ Tools and Software Version

- The entire project was created and developed in Windows 10 operating system.
- Programming Language : Python (3.7)
- Python IDE : Spider

Additional main libraries used:

- **Pandas**: Python has long been great for data managing and preparation, but less so for data analysis and modeling. Pandas is an open source library helps fill this gap, enabling you to carry out your entire data analysis workflow in Python.
  - To install type the following in Anaconda prompt (terminal):
    $ conda install pandas

- **scikit-learn**: Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. It's built upon some of the technology you might already be familiar with, like NumPy, pandas, and Matplotlib.
  - To install type the following in Anaconda prompt (terminal):
    $ conda install scikit-learn

- **Seaborn** is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
  - To install type the following in Anaconda prompt (terminal):
    $ **conda install Seaborn**

- **Plotly:** Its Python graphing library which makes interactive, publication-quality graphs.
  - To install type the following in Anaconda prompt (terminal):
    $ **conda install plotly**

❖ EDA and Preprocessing:

- While importing the data we got to know we had a total of 541909 observations and 8 features.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
InvoiceNo      541909 non-null object
StockCode      541909 non-null object
Description    540455 non-null object
Quantity       541909 non-null int64
InvoiceDate    541909 non-null datetime64[ns]
UnitPrice      541909 non-null float64
CustomerID     406829 non-null float64
Country        541909 non-null object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

- We checked for null values and found that CustomerID and Description had missing values.

```
missing_val(dataSet)
```

```
CustomerID     135080
Description      1454
Country             0
UnitPrice           0
InvoiceDate         0
Quantity            0
StockCode           0
InvoiceNo           0
dtype: int64
```

- We checked for duplicate values and found that there were 5225 observstions

```
dataSet.duplicated().sum()
```

```
5268
```

- We noticed that some observation corresponds to multiple orders for the same invoice and multiple invoices for the same customers as a result we checked the number of unique Customers and Invoices

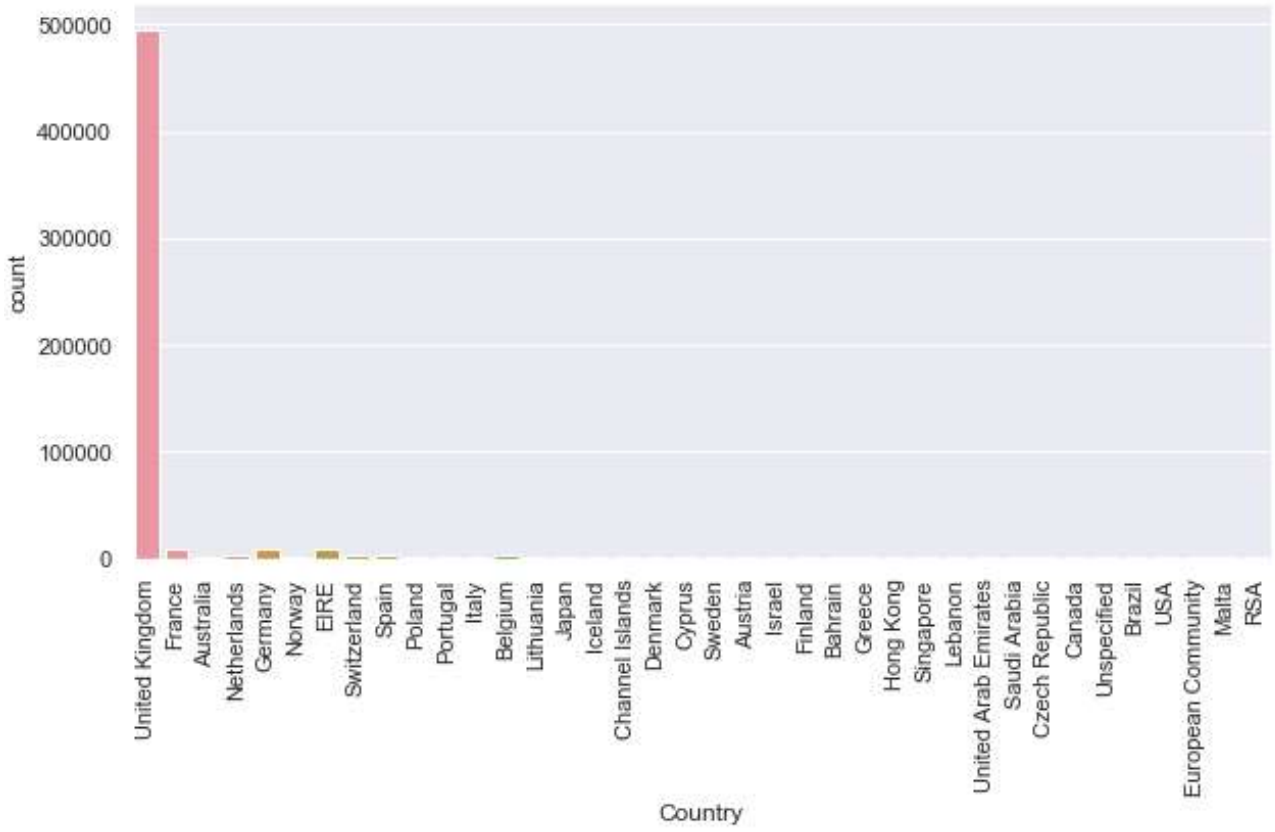
	Unique_Invoices	Unique_Customers
Count	25900	4372

- We also check the number of observations with negative values for quantities which corresponds to a ordered being cancalled.

```
cancelledOrder
```

```
9288
```

- We checked the total transactions made by customers from different countries and found that the majority of transactions were made by customers in the UK .



Data Cleaning:

- We decided to remove all the missing values from 'CustomerID' as it not advisable to impute an Identifier. This also resulted in removing null values in 'Description' also.
- As more than 90% of the transactions are made by customers from United Kingdom, we only considered those observations.
- We are deleting the columns 'StockCode' and 'Description' as they are not required for RFM analysis
- We removed the all the Cancelled orders and orders whose Unit Price is zero.
- We are only considering the Date for the 'InvoiceDate' Date-time Attribute(i.e. disregarding the time)

- Performing the above stated Data cleaning and manipulation steps resulted the following dataset.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 354321 entries, 0 to 541893
Data columns (total 5 columns):
CustomerID     354321 non-null float64
InvoiceDate    354321 non-null object
InvoiceNo      354321 non-null object
Quantity       354321 non-null int64
UnitPrice      354321 non-null float64
dtypes: float64(2), int64(1), object(2)
memory usage: 16.2+ MB
```

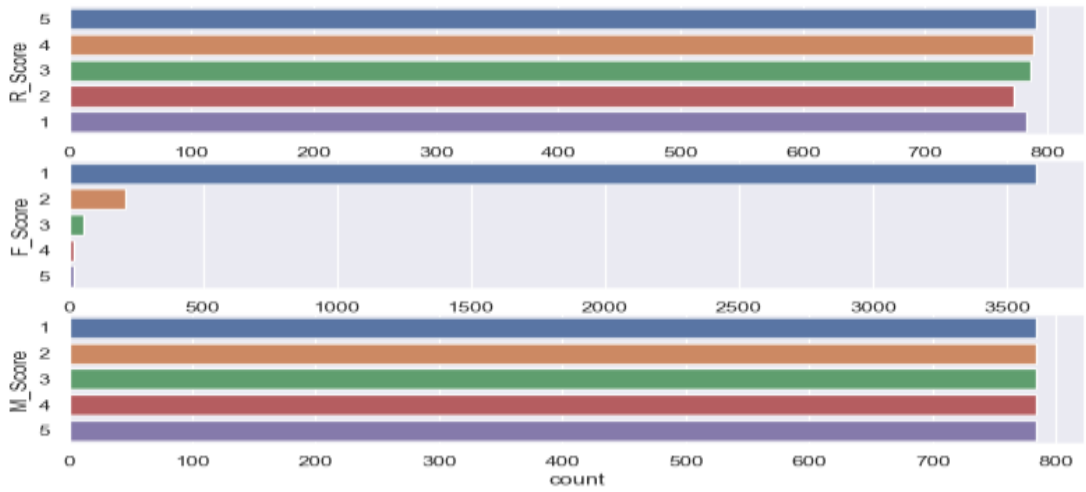
➢ Featuring Engineering:

- We created a new feature called ‘totalAmount’ by multiplying the 'UnitPrice' and 'Quantity' which states the final amount for each transaction.

❖ RFM analysis:

Further data preprocessing steps were performed stated below to implement RFM analysis.

- To calculate Recency, we grouped the data by CustomerID and considered the next day of the last InvoiceDate of purchase(i.e. 2011/12/10) and subtracted it from each date.
- To find how frequent a customer is, we grouped the data by CustomerID and added the number of unique Invoices in each group to obtain the total number of occurrences of purchase for a particular customer.
- Similarly, for Monetary value of a customer, we grouped the data by CustomerID and added the totalAmount for each group.
- We have assigned a ranking number of 1,2,3,4, or 5 (with 5 being highest) for each RFM parameter. We created a column ‘RFM_Cell’ combining three scores together where customers with ‘RFM_cell’ value being "555" are ideal. This resulted in giving us 3920 customers. Below is plot for count of R_Score, F_Score and M_Score.



- We have made another column 'RFM_Score' for easy reference and clustering purposes by taking the average of the RFM rankings(i.e. Scores)

```
rfm_df.head()
```

CustomerID	recency	frequency	monetary	R_Score	F_Score	M_Score	RFM_Cell	RFM_Score
12346.0	326	1	77183.60	1	1	5	115	2.333333
12747.0	3	11	4196.01	5	2	5	525	4.000000
12748.0	1	209	33719.73	5	5	5	555	5.000000
12749.0	4	5	4090.88	5	1	5	515	3.666667
12820.0	4	4	942.34	5	1	4	514	3.333333

❖ Clustering:

➢ **RFM Clustering:** For RFM clustering we have labeled the customer according to their RFM_Score into the following categories.

Segment	RFM Score	Description
Loyal and Best Customer	5	Highly engaged customers who have bought the most recent, the most often, and generated the most revenue
Potential Customers	4	Customers who buy the most often and recent
Promising and Faithfull Customers	3	Buyers you Can't-lose on your site and have moderate recency, frequency and monetary.
Customers at Risk	2	Customers who return often and spend from low to high range.
Already Lost Customers	1	Old customers who haven't bought in a while.

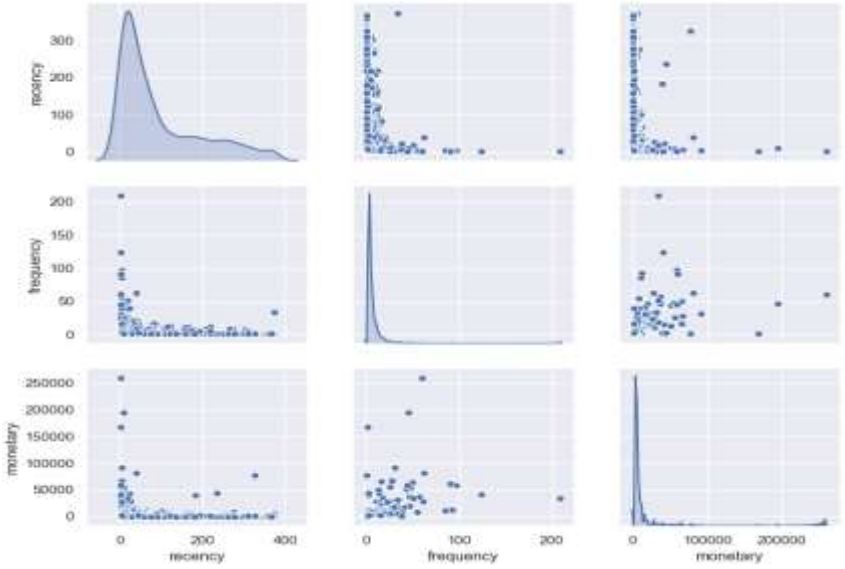
```
rfm_df['Customer Label'].value_counts()
```

```
Promising - Faithful Customers    1431
Customers at Risk                 1389
Potential Customer                 688
Already Lost Customers             325
Loyal and Best Customers            87
Name: Customer Label, dtype: int64
```

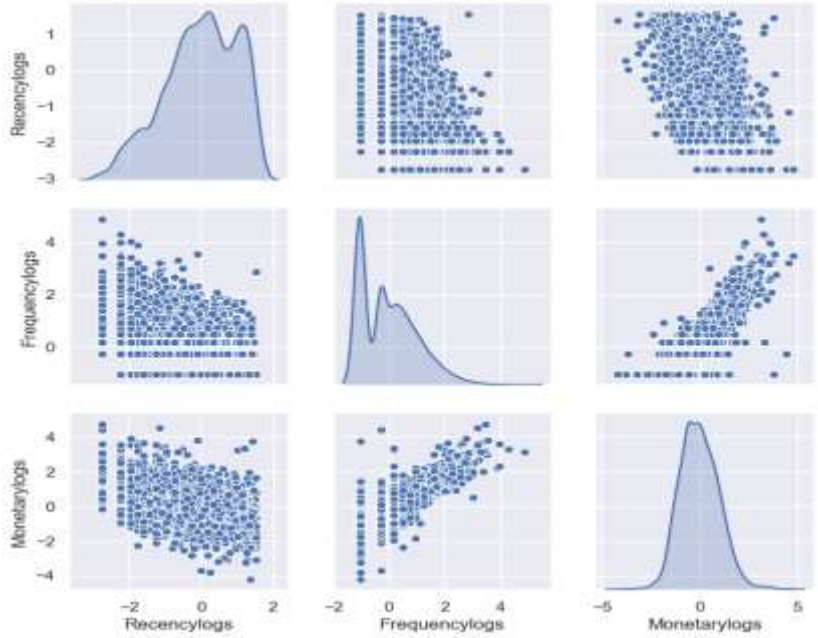
➢ **K-Means Clustering**:

Further data preprocessing steps were performed stated below to implement RFM analysis.

- We initially checked for outliers using Z-score and found that there were 70 observations from the 3920. Ideally these observations should be removed and analyzed separately but we decided to keep as they contributed towards 33.4 % of the entire monetary value and hence are very important
- We checked for skewness of and found that the data is skewed towards the right (figure 1) as a result we applied log trasformations to reduce positive skew and then standardize it.(figure 2).



(Figure:1 Before Log transformation)

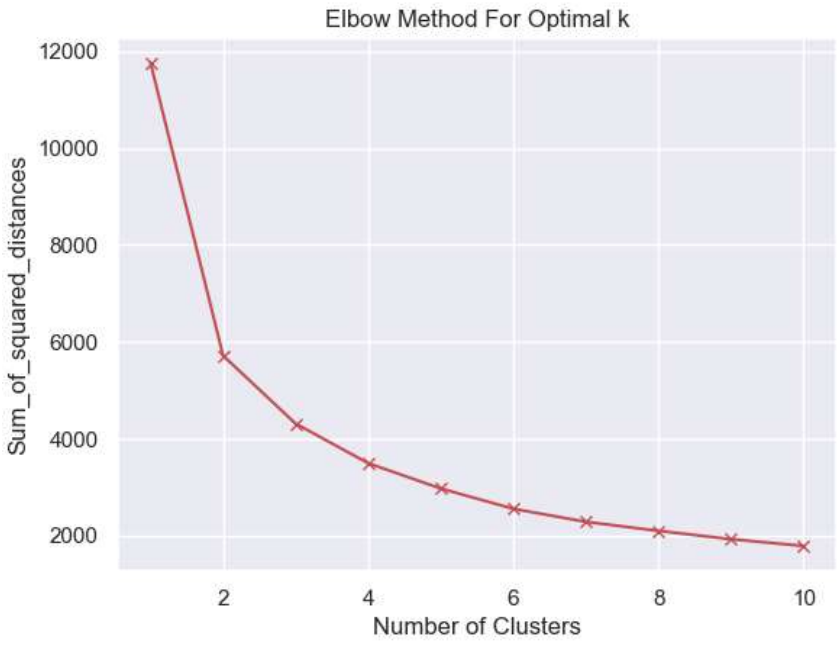


(Figure:2 After Log transformation)

```
RFMlog_s.describe()
```

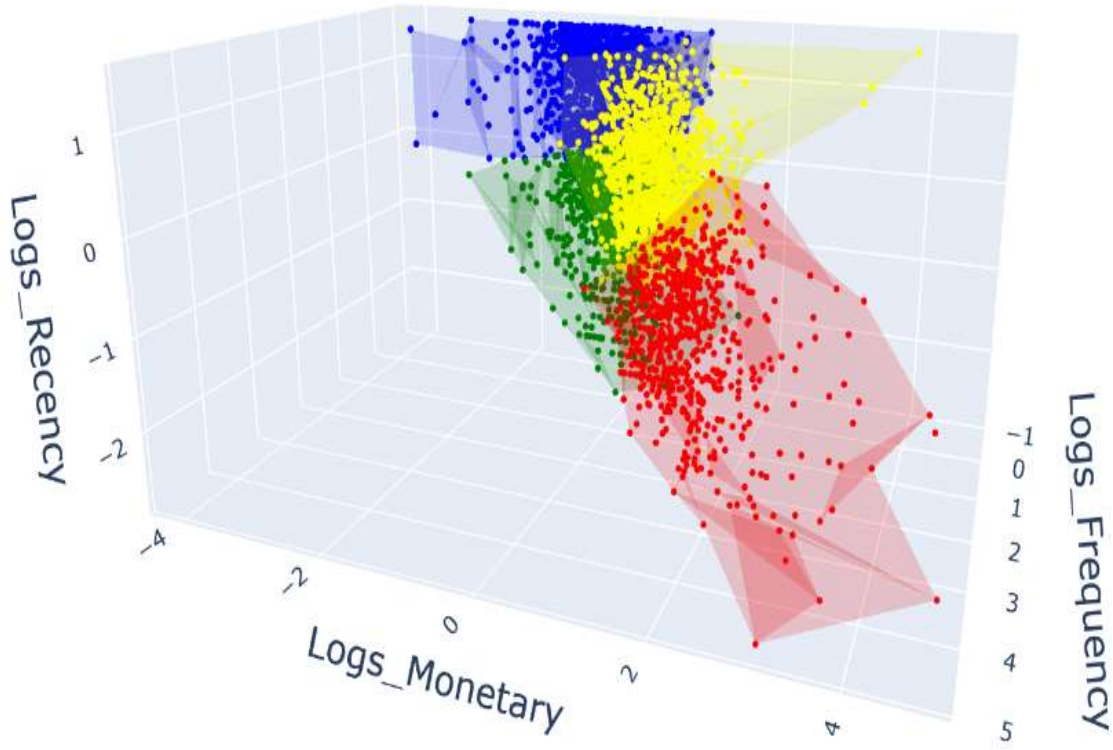
	Recencylogs	Frequencylogs	Monetarylogs
count	3.920000e+03	3.920000e+03	3.920000e+03
mean	3.987740e-17	1.586033e-17	1.377583e-16
std	1.000128e+00	1.000128e+00	1.000128e+00
min	-2.748403e+00	-1.050247e+00	-4.187313e+00
25%	-6.572134e-01	-1.050247e+00	-6.724217e-01
50%	9.628043e-02	-2.803087e-01	-5.030645e-02
75%	8.422246e-01	7.374949e-01	6.574437e-01
max	1.537807e+00	4.883947e+00	4.750651e+00

- In order find the value of 'k'(number of clusters) we have used the famous elbow method.(k=4)



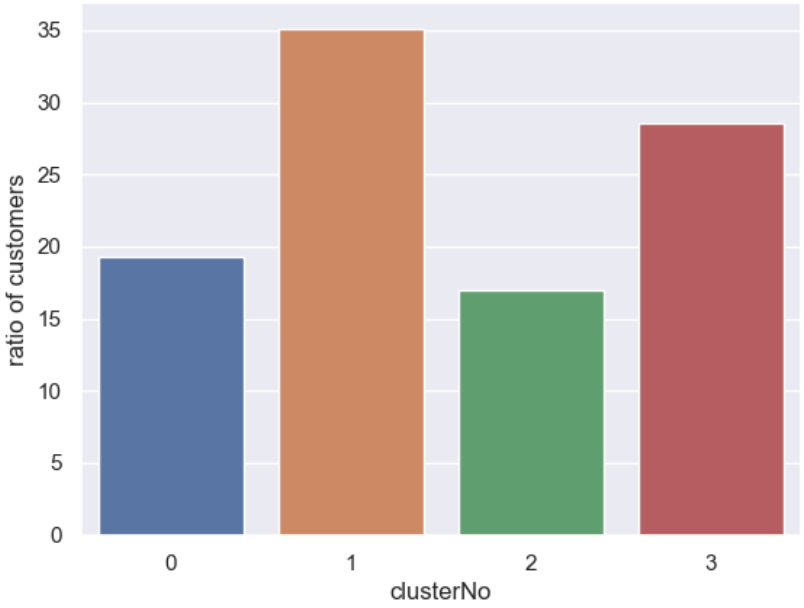
- We have implemented k-means using the following parameters and plotted the clusters in 3D plot

```
kmeans = KMeans(n_clusters = 4, init = 'k-means++',n_init=30)
```



(Figure:3)

- Ratio of customers in each cluster is shown in the plot below.



- Statistics of each cluster representing min, max, mean and median values of recency, frequency and monetary.

Cluster 0

	Mean	Minimum	Maximum	Median
recency	22.157199	1.0	64.0	20.00
frequency	1.911493	1.0	7.0	2.00
monetary	469.597635	30.0	3861.0	399.51

Cluster 1

	Mean	Minimum	Maximum	Median
recency	188.620915	31.00	374.00	185.00
frequency	1.273784	1.00	6.00	1.00
monetary	327.113290	3.75	2044.37	279.94

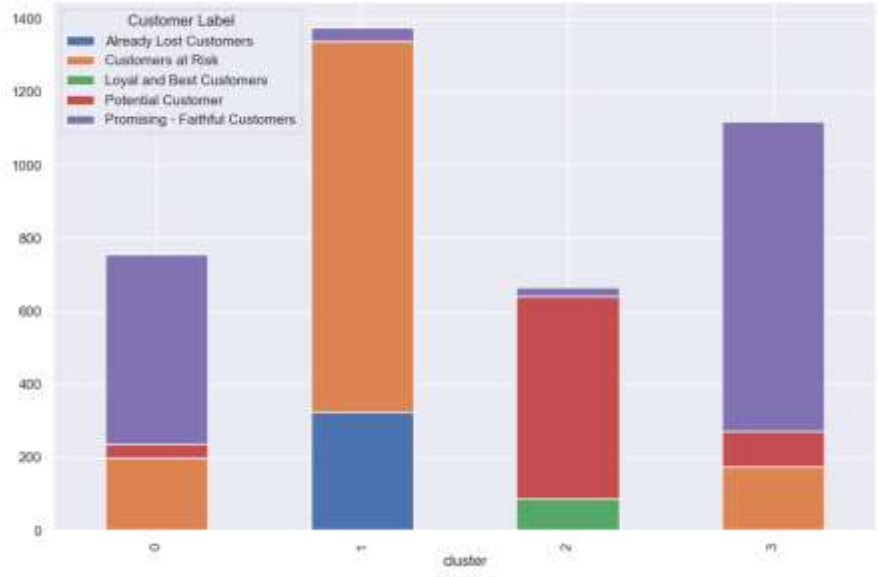
Cluster 2

	Mean	Minimum	Maximum	Median
recency	10.624625	1.00	90.0	8.000
frequency	13.054054	2.00	209.0	10.000
monetary	6947.573408	787.85	259657.3	3356.835

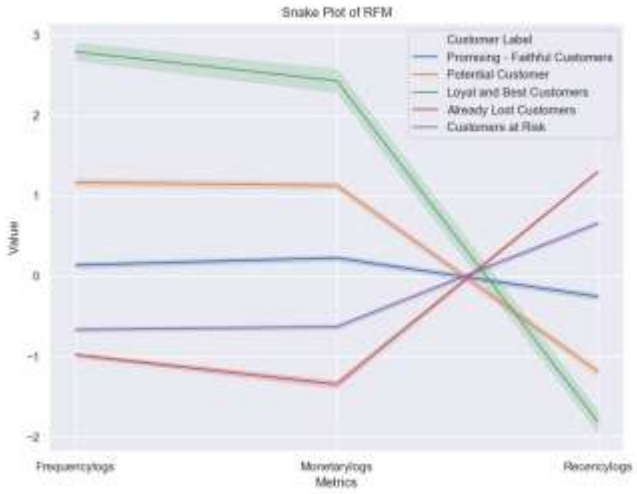
Cluster 3

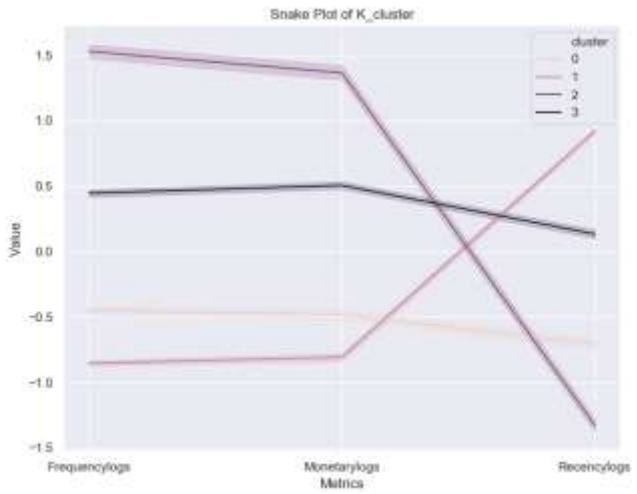
	Mean	Minimum	Maximum	Median
recency	71.400893	8.00	373.0	54.000
frequency	4.241964	1.00	34.0	4.000
monetary	1674.452904	226.75	77183.6	1246.045

- The distribution of different types of customers per RFM analysis in respective clusters is shown.



- A comaritive Snake plot is show below for both the RFM clustering and Kmeans respectively.





- From the above snake plots we state that loyal and potential groups on the left look similar to cluster 2, promosing group can relate to cluster 3 and the worst groups (Customers at risk and lost) to cluster 1.

➢ **Customer Analysis as per Cluster:**

- **Cluster 0**:- This cluster contains about 19% of the entire population and in a way can be considered a sub set of cluster 3. It is composed with a majority of promosing customers. The overall average RFM Score fro this cluster is arounnd 3

- **Cluster 1**: This cluster contains the most number of customers with a ratio of 35% of the entire populat--ion. It is composed of customers who are at risk and customers who are already at lost. The overall RFM score is less than 2.

- **Cluster 2**: This Cluster is composed of the most Profitable and Potential Customers. It contains about 17% of the whole population with the best RFM score.

- **Cluster 3**: This Cluster is composed of 29% of the entire population with a majority of promising and Faithful customers and a samll fraction of both potential and riskey customers. The overall RFM score is greater than 3 for this cluster.

❖ References:

1) https://link.springer.com/article/10.1057/dbm.2012.17#Fig3
2) https://www.sciencedirect.com/science/article/pii/S1319157818304178