Assignment 3

Unsupervised Learning Competition

REPORT

TABLE OF CONTENTS

TEAM: = (3 MEMBERS)

1. Junaid Charaniya (20169153)
2. Shashi Suman (20157041)
3. Vidhan Dholakia (20151326)

SOFTWARE USED:

We have used Python 3.7 for the coding purpose with the PyCharm IDE.

Instructions on how to Download & Install the software and the data sets:

How to Download Python & Install

Step 1: = Go to https://www.python.org/downloads/

Step 2: = Select our version and directly download as per the requirement and system adaptability.

Step 3: = The installation process is quite simple and straightforward. Just need to follow the obvious steps. We have used PyCharm IDE for execution of Python.

Installing Pandas, Numpy, Keras, Matplotlib:

Pandas:

Install Python on your PC. Open the command prompt and type

pip install pandas

Numpy and Matplotlib can be similarly installed by using the following commands:

pip install numpy

pip install Matplotlib

For Seaborn, we need to type the normal command:

pip install seaborn

For Sklearn, we need to type the command:

pip install Sklearn

How to Download the Data Set?

We have been given the data set of the online shoppers of the retail sales. Over a decade, there has been a steady and strong increase of online retail sales. According to the Interactive Media in Retail Group (IMRG), online shoppers in the United Kingdom spent an estimated £50 billion in year 2011, a more than 5000 per cent increase compared with year 2000. This remarkable increase of online sales indicates that the way consumers shop for and use financial services has fundamentally changed.

ANALYSIS PROCESS USED:

We have the data about the online retail of the shopper in the UK which compares the rise in the sales over a decade (2000 to 2010). The data set contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.

Our task is to create clusters and segment the customers into various groups using the clustering algorithm based on the RFM analysis. After segmenting, the task is to identify the vital features and characteristics of the consumers in each segment.

Initially, we had the total data of 541910 items! This data set is not useful yet as it is not clean, and it is totally unfit for the data modelling processes.

voiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING H	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LAN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEA	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FL	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTT	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA N	2	12/1/2010 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROST	6	12/1/2010 8:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UN	6	12/1/2010 8:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RE	6	12/1/2010 8:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUF	32	12/1/2010 8:34	1.69	13047	United Kingdom
536367	22745	POPPY'S PLAYHOUS	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	22748	POPPY'S PLAYHOUS	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	22749	FELTCRAFT PRINCE	8	12/1/2010 8:34	3.75	13047	United Kingdom
536367	22310	IVORY KNITTED MU	6	12/1/2010 8:34	1.65	13047	United Kingdom
536367	84969	BOX OF 6 ASSORTE	6	12/1/2010 8:34	4.25	13047	United Kingdom
536367	22623	BOX OF VINTAGE JI	3	12/1/2010 8:34	4.95	13047	United Kingdom
536367	22622	BOX OF VINTAGE A	2	12/1/2010 8:34	9.95	13047	United Kingdom
536367	21754	HOME BUILDING BL	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	21755	LOVE BUILDING BL	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	21777	RECIPE BOX WITH M	4	12/1/2010 8:34	7.95	13047	United Kingdom
536367	48187	DOORMAT NEW EN	4	12/1/2010 8:34	7.95	13047	United Kingdom
536368	22960	JAM MAKING SET V	6	12/1/2010 8:34	4.25	13047	United Kingdom
536368	22913	RED COAT RACK PA	3	12/1/2010 8:34	4.95	13047	United Kingdom

Fig. Data (Unclean)

We have to clean this data in order to apply the clustering algorithm.

DATA PREPARATION & DATA CLEANING: =

We have updated our data as per our requirements. Firstly, we have cleaned the data by dealing with the missing values and bifurcating between the relevant and irrelevant features. Microsoft Excel was used for data cleaning. Python was used to apply k means algorithm on the cleaned data.

For instance, we removed the "COUNTRY, DESCRIPTION, STOCK CODE" column for applying the algorithm. This factor was totally extraneous for us and we will remove it directly.

Our main task is to calculate the Recency, Frequency and monetary for the customer and then apply the clustering algorithm. For this, we need the "INVOICE DATE" and essentially "CUSTOMER ID". We have merged the total number of items (QUANTITY) and UNIT PRICE to get the total amount.

There were records where CUSTOMER ID were not present. We removed those records as well

Now, we have cleaned, and updated data as shown below:

Customer	Recency	Frequency	Total Amc	R	F	M	RMF Score
12346	325	1	77183.6	1	1	5	7
12347	2	182	4310	5	5	5	15
12348	75	31	1797.24	2	3	4	9
12349	18	73	1757.55	4	4	4	12
12350	310	17	334.4	1	2	2	5
12352	36	85	2506.04	3	4	5	12
12353	204	4	89	1	1	1	3
12354	232	58	1079.4	1	4	4	9
12355	214	13	459.4	1	1	2	4
12356	22	59	2811.43	4	4	5	13
12357	33	131	6207.67	3	5	5	13
12358	1	19	1168.06	5	2	4	11
12359	57	248	6372.58	3	5	5	13
12360	52	129	2662.06	3	5	5	13
12361	287	10	189.9	1	1	1	3
12362	3	266	5226.23	5	5	5	15
12363	109	23	552	2	2	3	7
12364	7	85	1313.1	5	4	4	13
12365	291	22	641.38	1	2	3	6
12367	4	11	168.9	5	1	1	7
12370	51	167	3545.69	3	5	5	13
12371	44	63	1887.96	3	4	4	11
12372	71	52	1298.04	3	3	4	10
12373	311	14	364.6	1	2	2	5

Fig. Data (clean)

Now that we have the cleaned data, we must apply the RFM analysis on the data before applying the clustering algorithm.

RFM ANALYSIS:

We have calculated the recency, frequency and monetary for every customer. For the classification of the 3 values, we have divided the total number of values by 5. After doing this, we will assign a number (from 1 to 5) in order to rank the customer (category wise). The lower is the frequency, lower will be the rank of F. The lower the recent, higher will be rank of R. The lower the monetary value, the lower will be the rank of M. We first went ahead with equal width binning which resulted in a poor cluster formation. Hence, we went ahead with equal density binning. For example, thus there must approximately same number of customers with frequency (F) rank 1, 2, 3, 4 as well as 5. Similarly, equal density binning were applied for Monetary (M) and Recency(R) as well.

Calculation of RFM:

Recency:

We have calculated the recency by taking the difference of the last order and the last date of the study which in this case turns out to be (09/12/2011).

For our data, we have calculated the recency for every customer. The customer with the maximum recency value is basically considered to have visited a very long time ago. In other words, if recency has a high value, that customer is not recent and has not visited the store in recent days. If a customer visits the store on ever day basis, the value is low.

For example, a customer with 300 recency value has not visited the store recently for the past 300 days. A customer with 1 recency value has visited the store recently **i.e** the previous day.

Frequency:

We have calculated the frequency by counting the transactions per customer. The more a customer visits the store, the more the frequency count will be.

For our data, we have calculated the frequency for every customer. In this case, if the frequency count is low, the customer is not frequent enough in visiting the store.

For example, a customer with 1 frequency value is less frequent whilst a customer with 300 frequency value has visited the store a greater number of times.

Monetary:

We have calculated the monetary by multiplying the total number of quantities with the unit price. This gives us the total cost or the monetary for every customer. More the amount, more is the monetary value.

APPROACHES USED AND ACCURACY:

We have to apply the clustering data mining algorithm to create the cluster and identify/address certain assumptions and address the issues of the shoppers regarding the online sales.

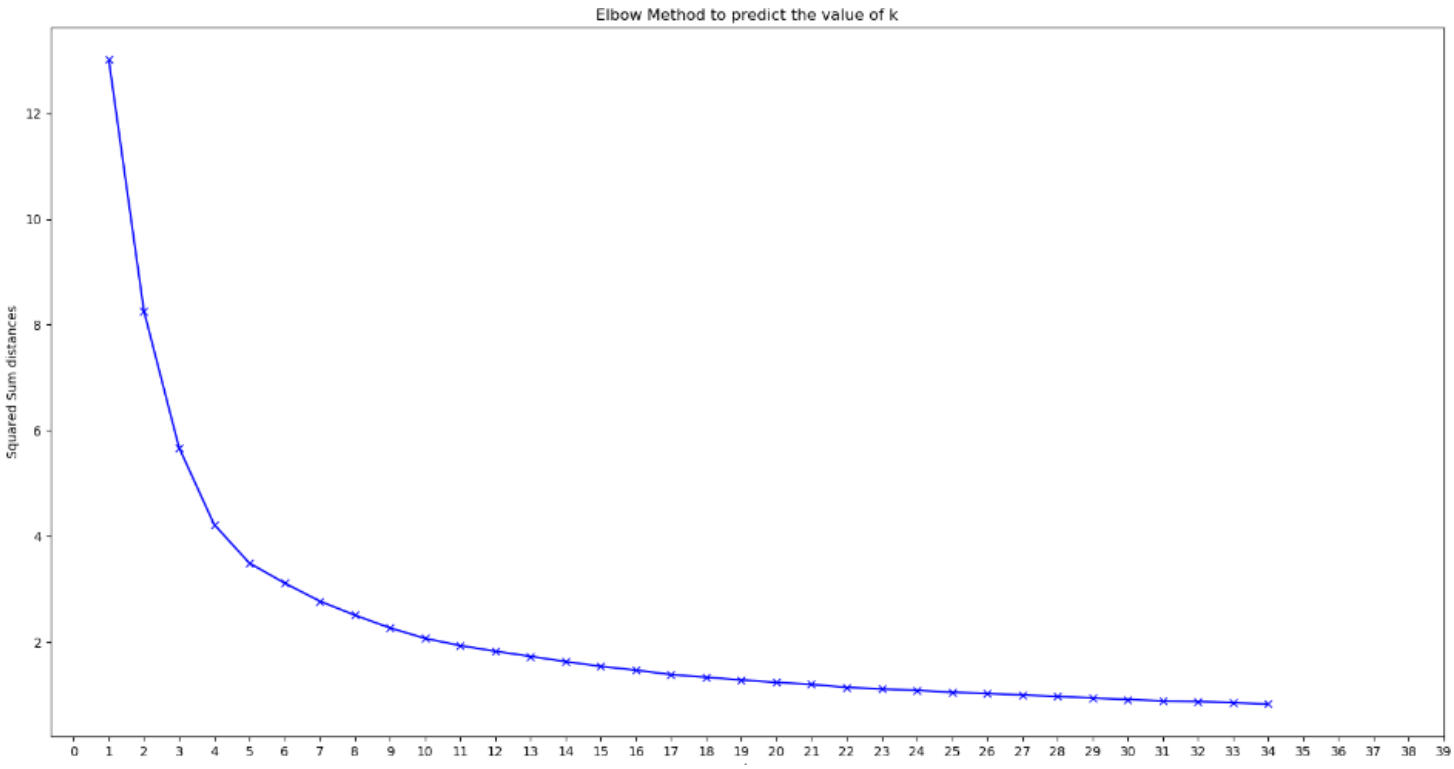
We have applied the K-Means clustering algorithm on the data we have developed.

We got to find the optimal value of k in KMEANS CLUSTERING algorithm for which we have used the Elbow method. The Elbow method is one of the most common methods to determine the optimal value of K.

The idea of this method is to run K-means clustering on the data set of a range of values of k and for each value of k, we calculate the sum of squared errors (SSE). We have then plotted a line chart of the SSE for each value of k. If the line chart looks like an arm, then the elbow on the arm is the value of k (which is expected to be the best). The main idea is that we want and expect a small SSE, but the catch is the SSE tends to decrease toward 0 as we increase the value of k.

Our goal is to choose a smaller value of k that still has a low SSE, and the elbow usually represents where we start to have diminished returns by increasing the value of k.

Note: The elbow method does not work well if the data is not very clustered.



We have applied K-Means clustering algorithm on the data and we get following results.

K = 4 (4 Clusters)

The number of times, slope of the elbow graph falls, a cluster is present at that instance. As we can see from the figures below, we have formed 4 clusters with the statistical measures (mean, median and max values) for Recency, Frequency and Monetary.

For the accuracy, we will measure the Silhouette score for every value of k. The closer the value is to zero, the more accurate the cluster is formed.

A Silhouette score helps to predict the decision boundary. Higher value indicates neighbors are far away from each other. Low value indicates that clusters are near each other.

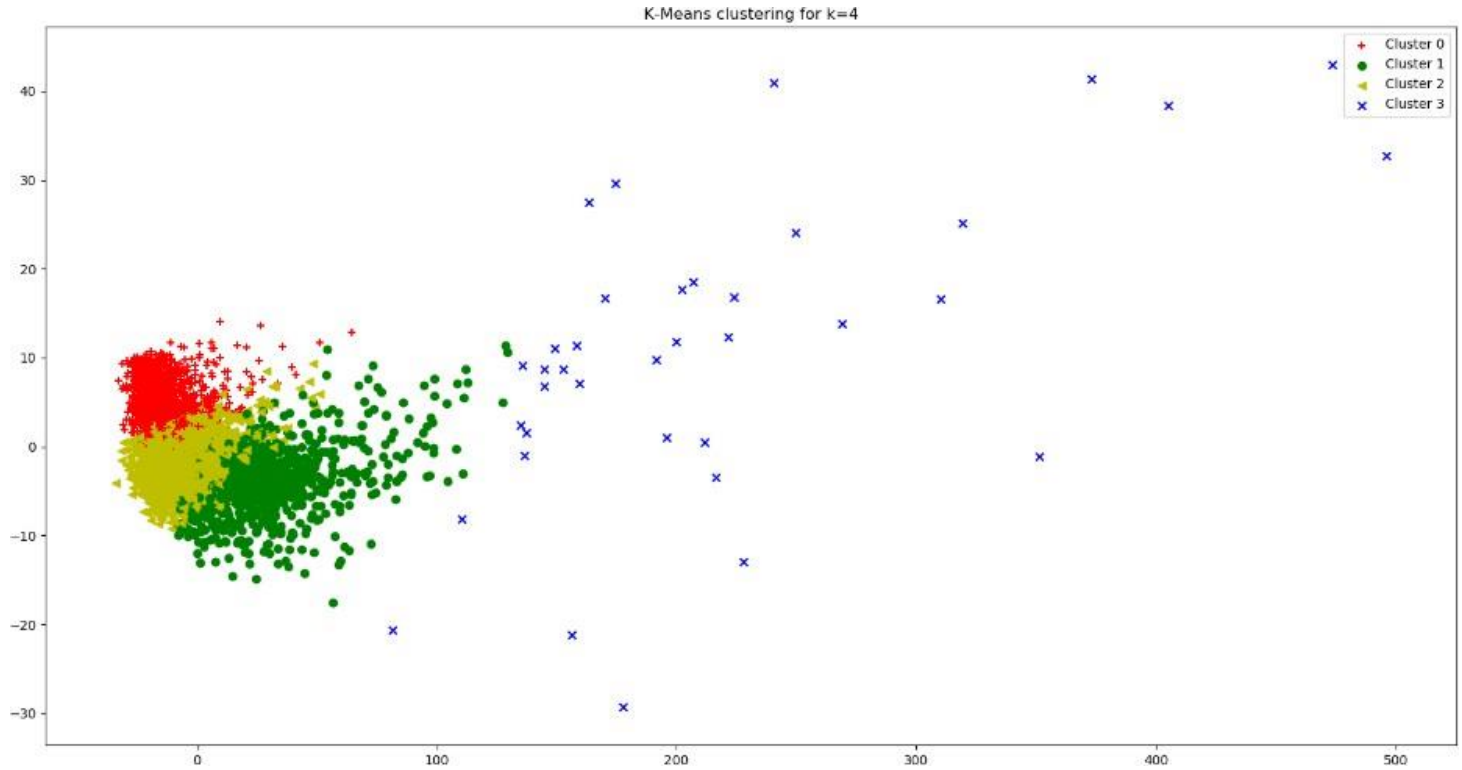
Code:

```
preds = kmean.fit_predict(dt)

score = silhouette_score(dt, preds, metric='euclidean')
print("For 6 clusters the silhouette is", score)
```

For 4 clusters the silhouette score is 0.41411414710812144

```
C:\Users\18jc137\PycharmProjects\Project_1\venv\Scripts\python.exe C:/Users/18jc137/PycharmProjects/Project_1/Comp_3.py
            Recency                Cluster
               mean median  max   count
Cluster
0         227.360000    218  373    1275
1          22.521437     14  275     863
2          41.162200     35  138    2164
3          26.207297      4  325      37
           Frequency                Cluster
                mean median   max   count
Cluster
0          27.483922     19   297    1275
1         250.485516    199  1677     863
2          46.843808     36   192    2164
3        1225.486486    431  7847      37
         Total Amount                          Cluster
                 mean     median       max   count
Cluster
0          498.181279    316.580    9864.26    1275
1         4369.834148   3212.840   26879.04     863
2          843.476226    641.535    7330.80    2164
3        72426.414865  51527.300  280206.02      37
```



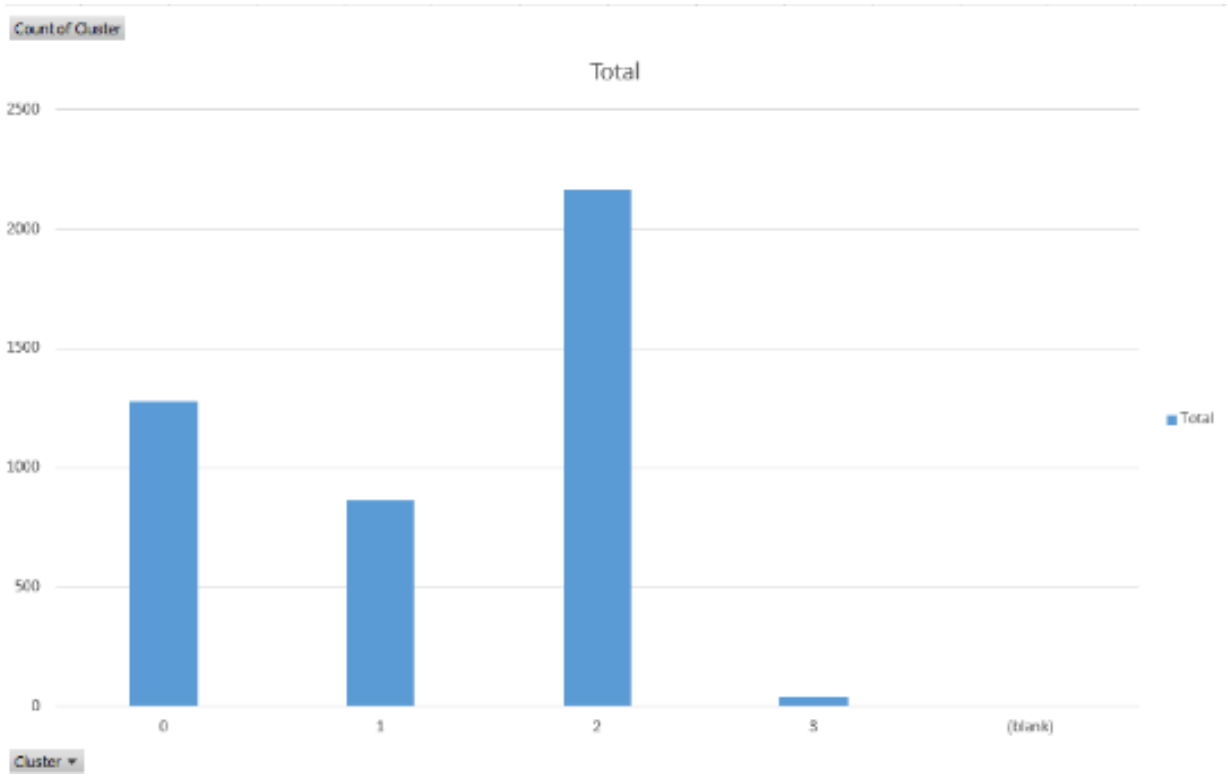


Fig. Count of Clusters

Row Labels	Count of Cluster
0	1275
1	863
2	2164
3	37
(blank)	
Grand Total	**4339**

K = 5 (5 Clusters)

As we can see from the figures below, we have formed 5 clusters with the statistical measures (mean, median and max values) for Recency, Frequency and Monetary.

For 5 clusters the silhouette score is 0.37288342306951683

```
Comp_3
C:\Users\18jc137\PycharmProjects\Project_1\venv\Scripts\python.exe C:/Users/18jc137/PycharmProjects/Project_1/Comp_3.py
            Recency            Cluster
               mean median  max   count
Cluster
0         48.042027     45  122    1618
1        235.112322    232  373    1193
2         15.171429      9  235     350
3         20.610727     21  211    1156
4         20.500000      3  325      22
             Frequency              Cluster
                  mean median   max   count
Cluster
0          29.868974   27.0   100    1618
1          27.020823   19.0   297    1193
2         389.954286  333.5  2700     350
3         124.216263  111.0   378    1156
4        1694.318182  652.5  7847      22
             Total Amount                        Cluster
                    mean     median       max   count
Cluster
0          578.075935    447.310    4507.01    1618
1          478.610839    312.380    7832.47    1193
2         7977.491543   5657.985   44534.30     350
3         2085.575926   1792.365   12393.70    1156
4       100834.368182  65980.175  280206.02      22
0 -2.486568653986987

Process finished with exit code 0
```

Fig Clusters with their statistical measures

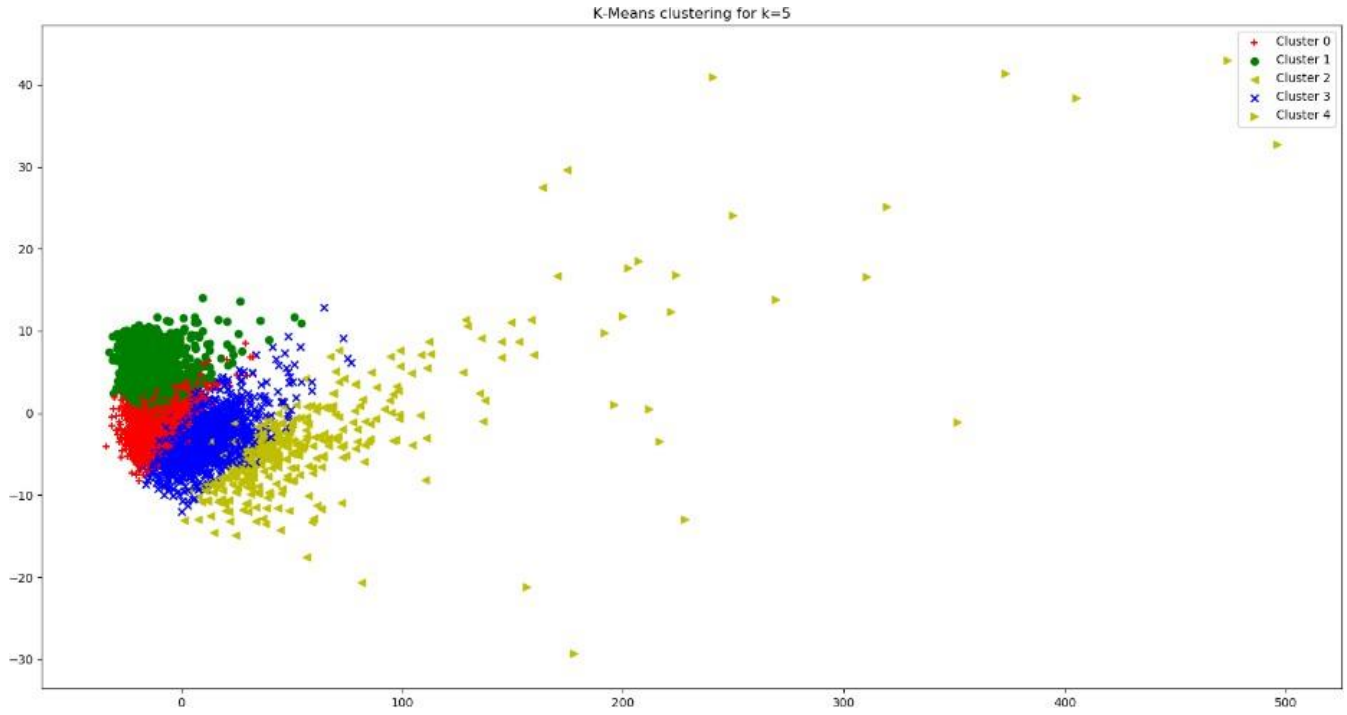


Fig Cluster Plot for K= 5

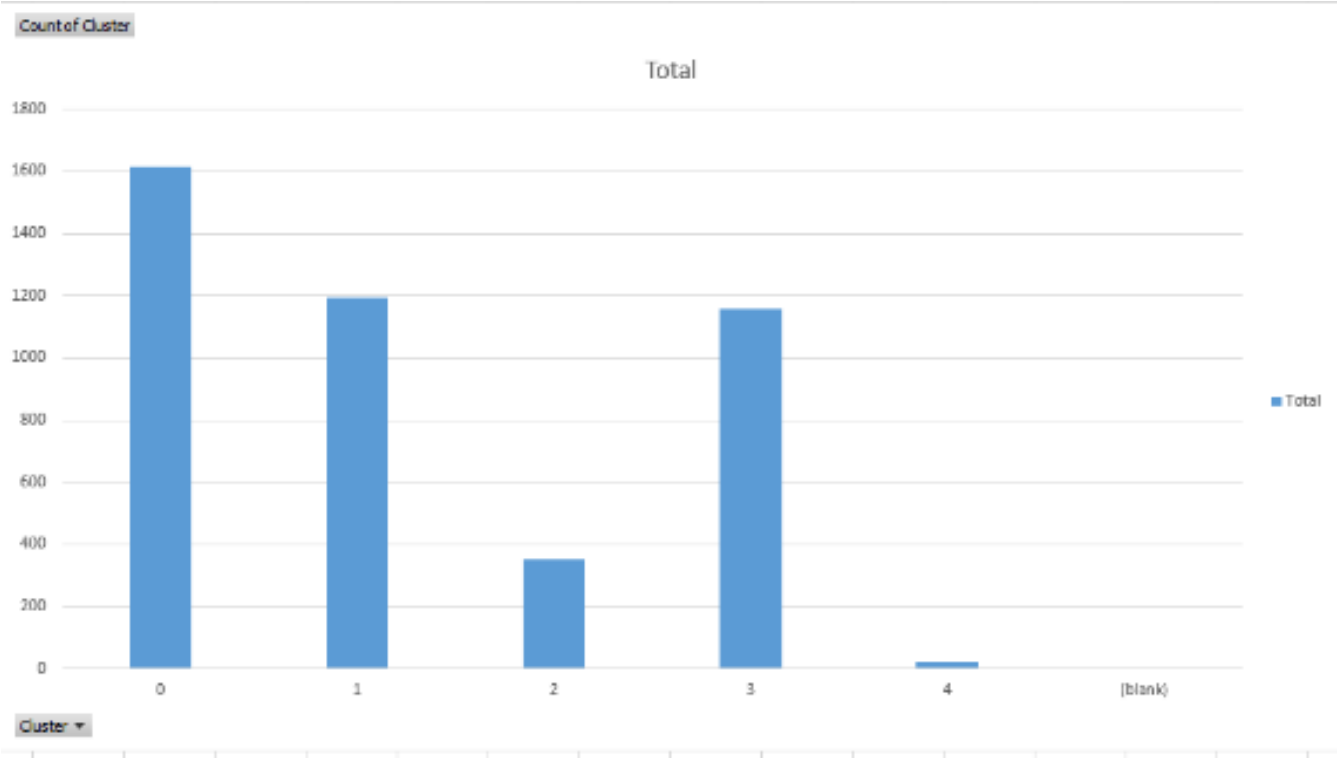


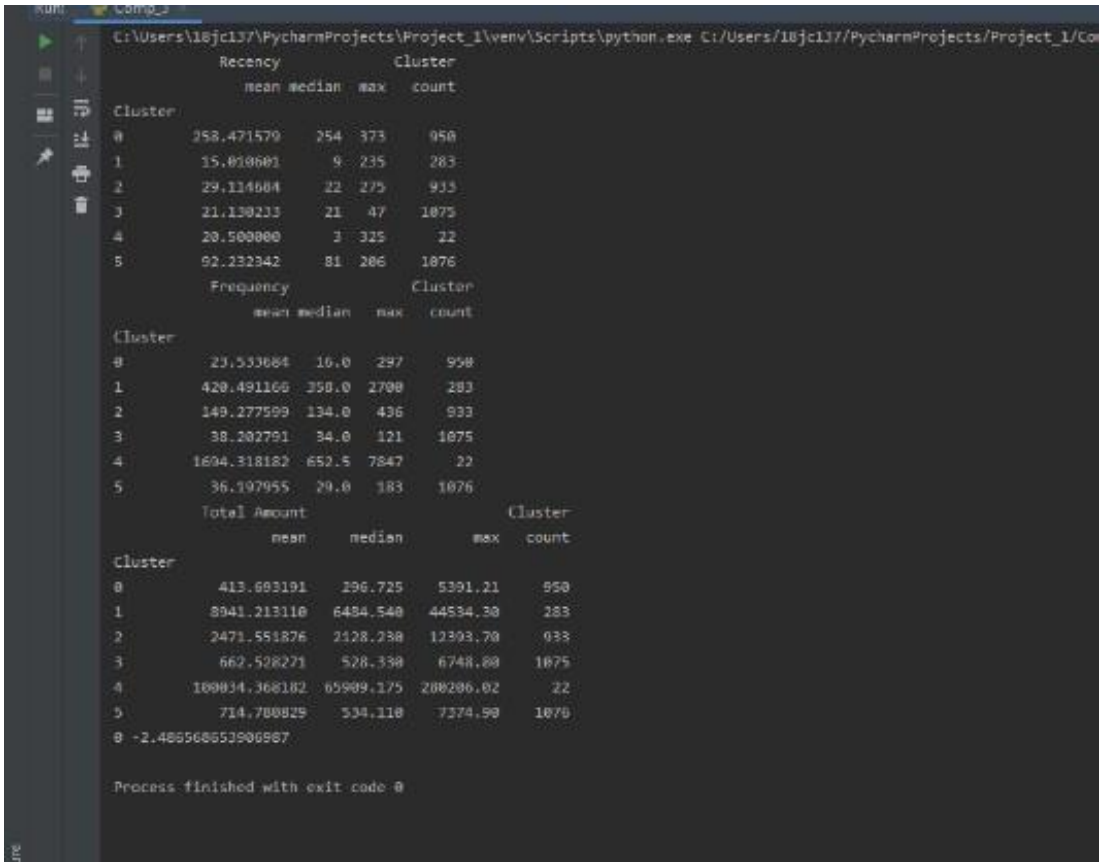
Fig Count of Cluster for K = 5

	A	B	C
1			
2			
3	Row Labels	Count of Cluster	
4	0	1618	
5	1	1193	
6	2	350	
7	3	1156	
8	4	22	
9	(blank)		
10	Grand Total	4339	
11			
12			
13			

K = 6 (Six Clusters)

As we can see from the figures below, we have formed 6 clusters with the statistical measures (mean, median and max values) for Recency, Frequency and Monetary.

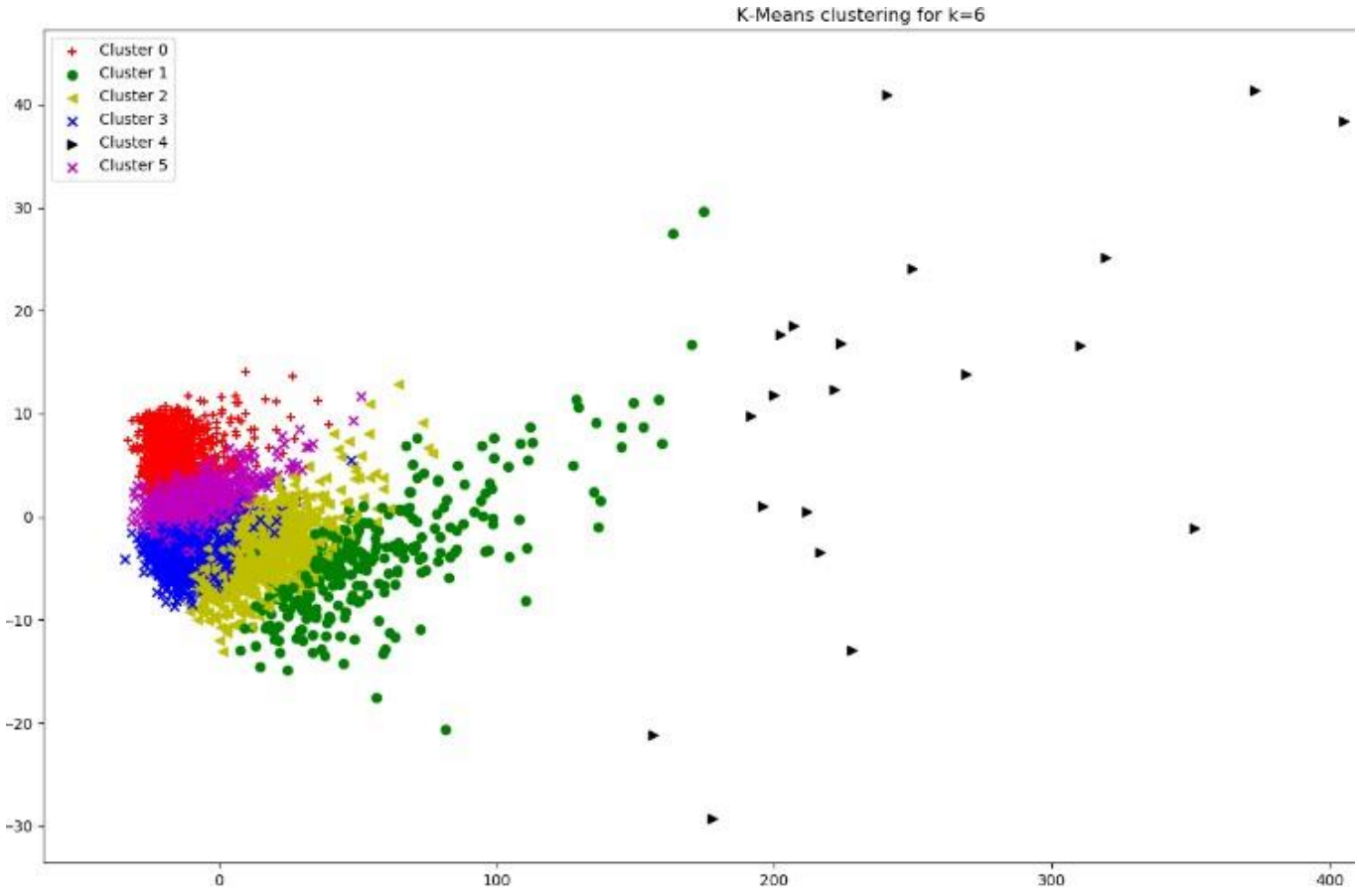
For 6 clusters the silhouette score is 0.32227566755106996

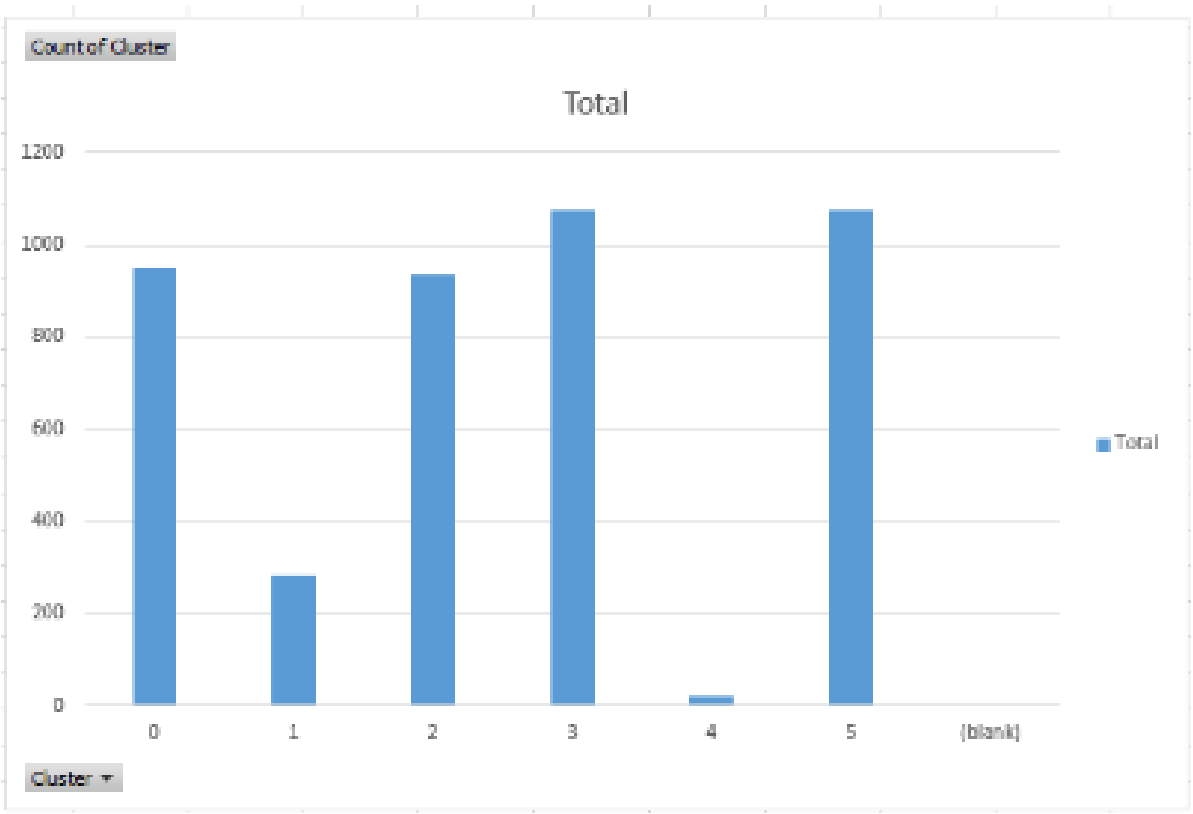


```
C:\Users\18jc137\PycharmProjects\Project_1\venv\Scripts\python.exe C:/Users/18jc137/PycharmProjects/Project_1/Co
            Recency            Cluster
               mean median  max   count
Cluster
0         258.471579    254  373     950
1          15.010601      9  235     283
2          29.114684     22  275     933
3          21.130233     21   47    1075
4          20.500000      3  325      22
5          92.232342     81  286    1076
           Frequency             Cluster
                mean median   max   count
Cluster
0          23.533684   16.0   297     950
1         420.491166  350.0  2700     283
2         149.277599  134.0   436     933
3          38.202791   34.0   121    1075
4        1604.318182  652.5  7847      22
5          36.197955   29.0   183    1076
          Total Amount                       Cluster
                  mean     median        max   count
Cluster
0           413.603191    296.725    5391.21     950
1          8941.213110   6484.540   44534.30     283
2          2471.551876   2128.230   12393.70     933
3           662.528271    528.330    6748.80    1075
4        100034.368182  65909.175  200206.02      22
5           714.780829    534.110    7374.90    1076
0 -2.405568653906987

Process finished with exit code 0
```

Fig Six Cluster with statistical measures


K-Means clustering for k=6
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
40
30
20
10
0
−10
−20
−30
0
100
200
300
400

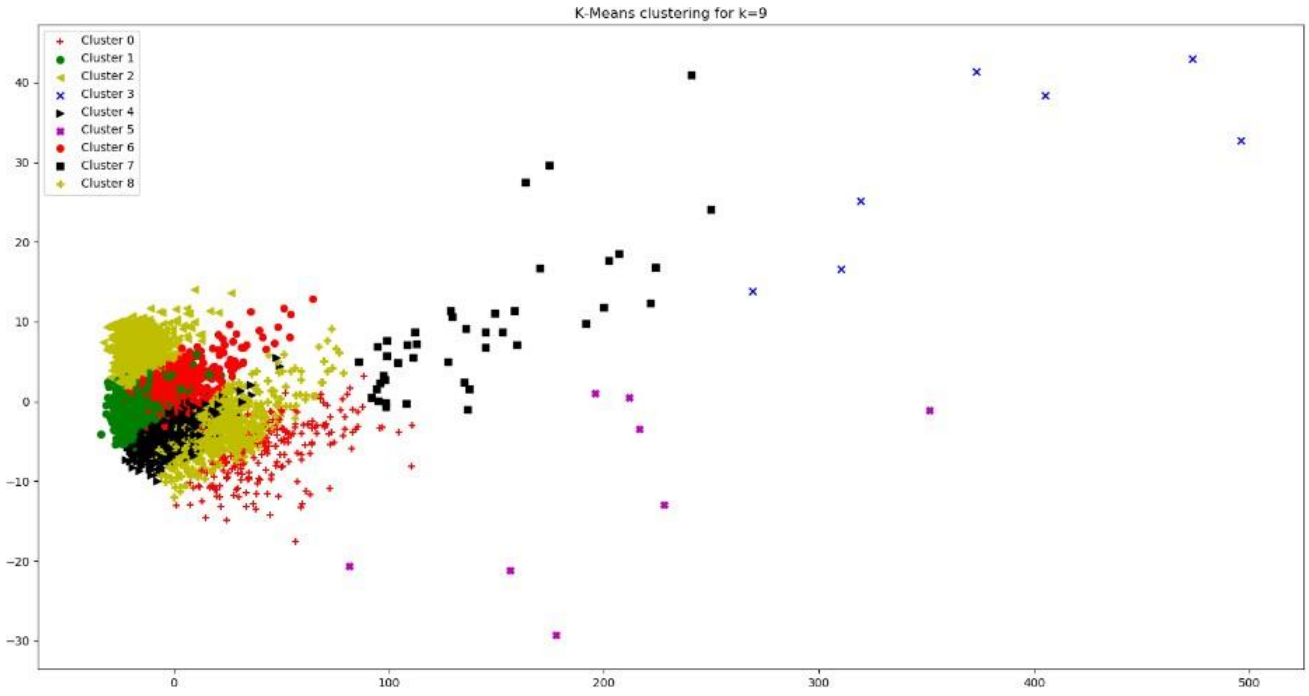


	A	B	C
1			
2			
3	**Row Labels**	**Count of Cluster**	
4	0	950	
5	1	283	
6	2	933	
7	3	1075	
8	4	22	
9	5	1076	
10	(blank)		
11	**Grand Total**	**4339**	
12			
13			

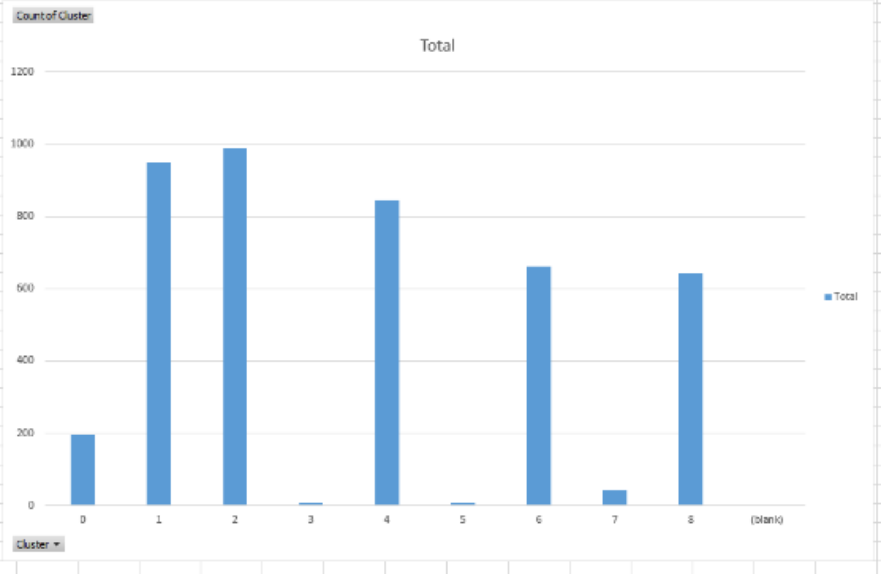
K = 9 (Nine Clusters)

We also tried extending the value of K = 9 but we found out that if we create 9 clusters, the data gets totally distorted and biased.

For 9 clusters the silhouette score is 0.32504673620871455



Row Labels	Count of Cluster
0	197
1	949
2	988
3	7
4	845
5	8
6	661
7	42
8	642
(blank)	
Grand Total	4339



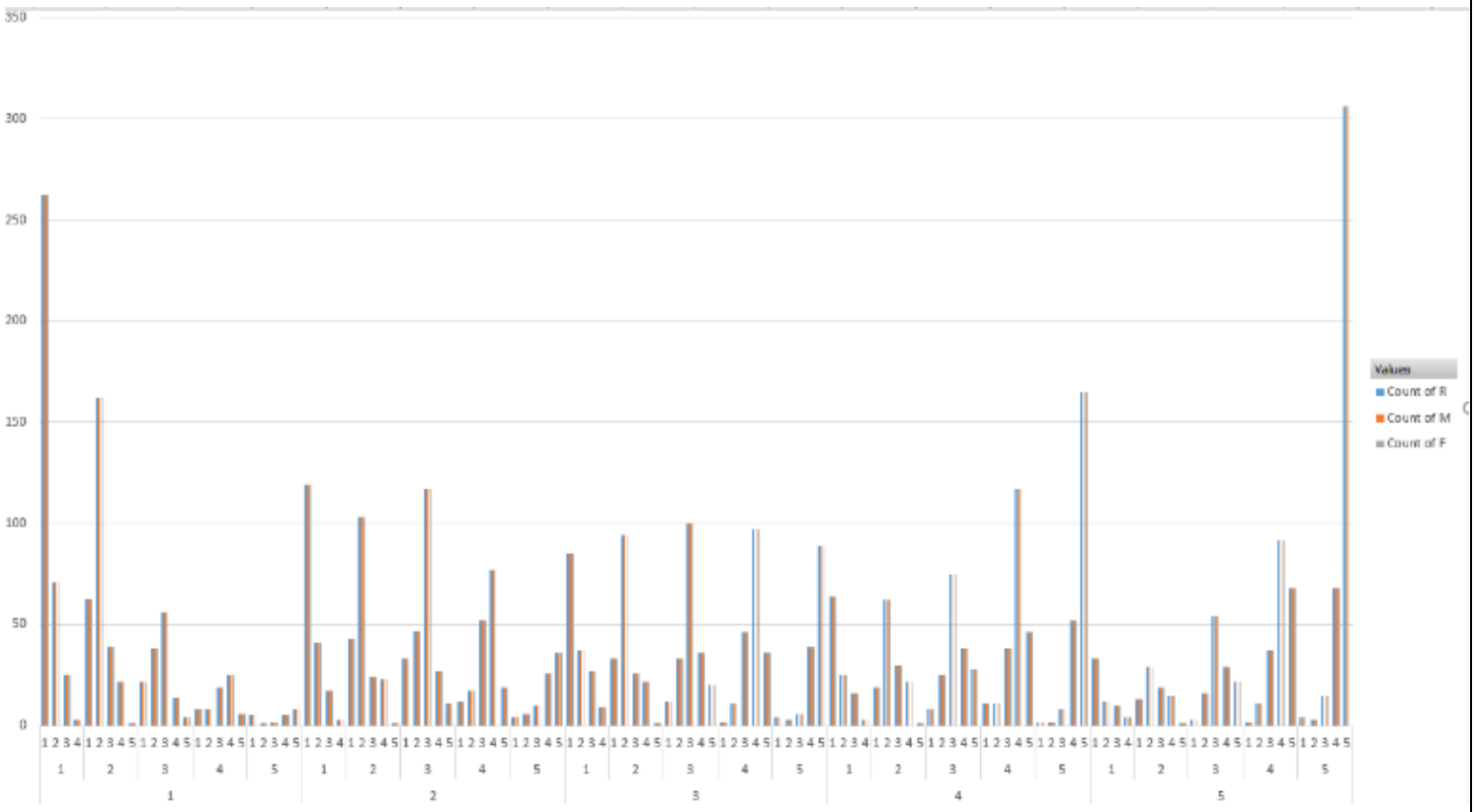


Fig RFM Distribution for all permutations of RFM

The lowest row is value of R. The second lowest row is F and the top row is M. We are basically showing population of all permutations of RFM like for 111, 112, 113, 114, 115, 211, 212, 213, 214, 215, 311……

REFERENCES:

[1] https://www.python.org/doc/
[2] https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
[3] https://www.datacamp.com/community/tutorials/introduction-customer-segmentation-python