

REPORT

TABLE OF CONTENTS

[Team: = (3 members) 2](#_bookmark0)

[Software Used: 2](#_bookmark1)

[Analysis Process Used: 4](#_bookmark2)

[Data Preparation & Data Cleaning: = 5](#_bookmark3)

[Approaches Used and Accuracy 7](#_bookmark4)

[References: 18](#_bookmark5)

**1**

# TEAM: = (3 MEMBERS)

|  |  |
| --- | --- |
| 1. 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.

**2**

**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.

**3**

# 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.

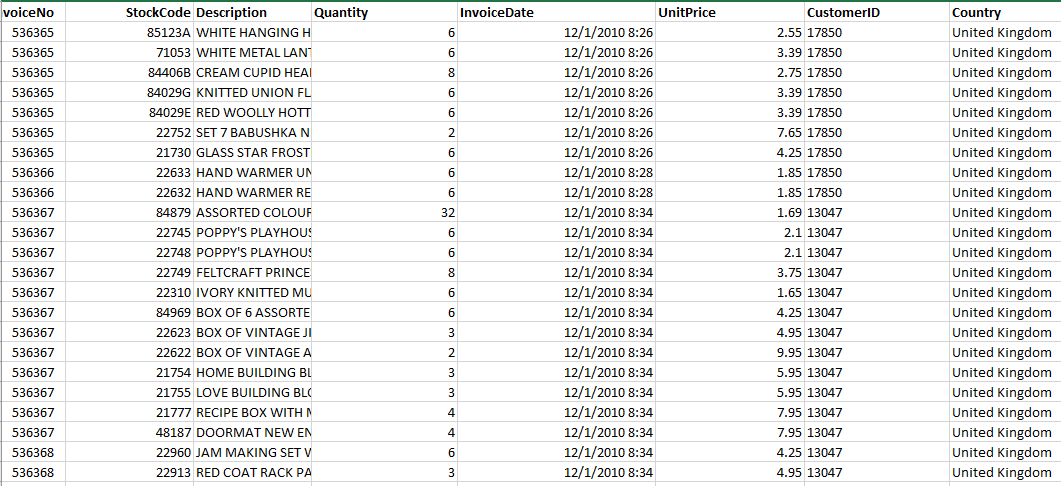


Fig. Data (Unclean)

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

**4**

# 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:

**5**

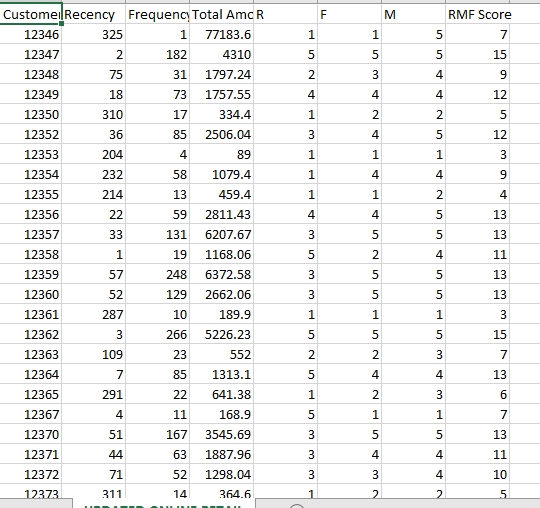


Fig. Data (clean)

**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:**

**6**

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

* 1. 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.

**7**

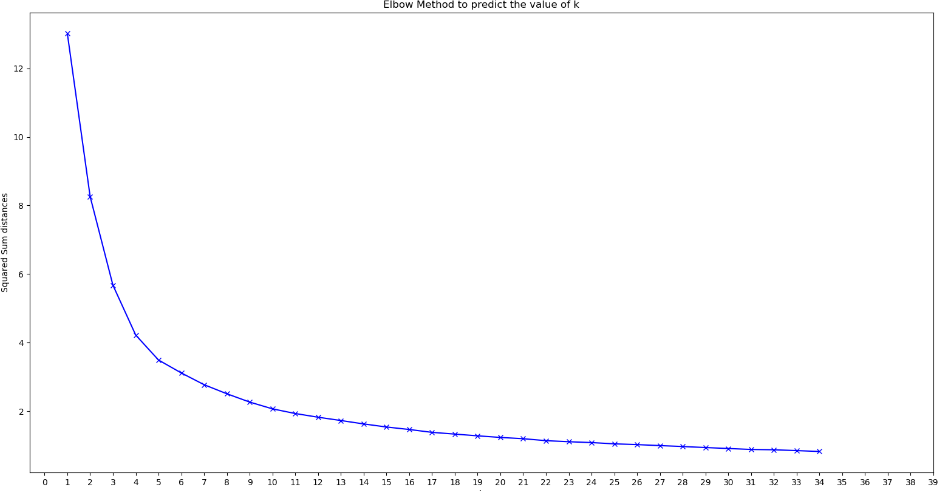
We have applied the K-Means clustering algorithm on the data we have developed.

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.

**8**

**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.**

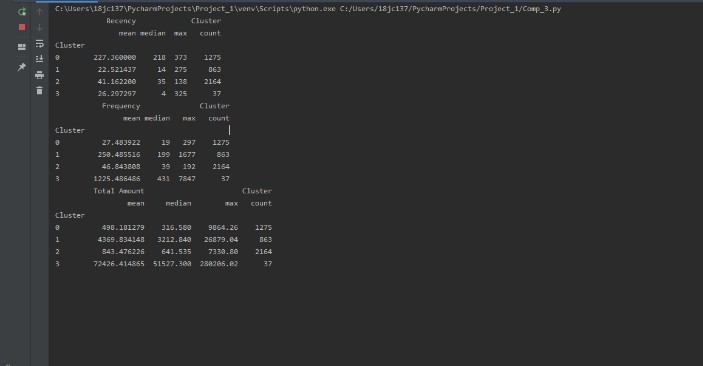
**Code:**

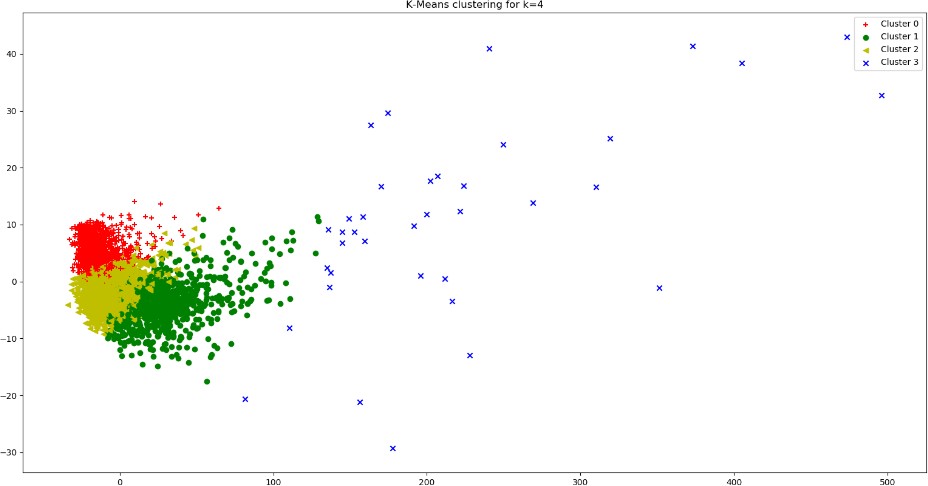
preds = kmean.fit\_predict(dt)

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

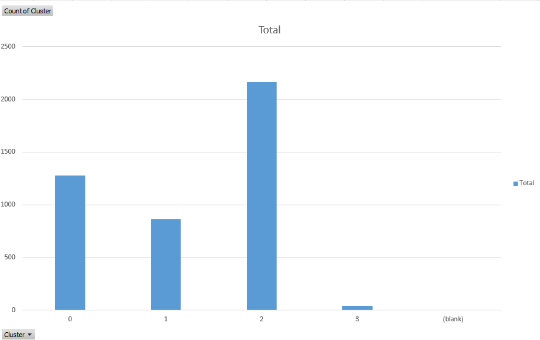
For 4 clusters the silhouette score is 0.41411414710812144

**9**



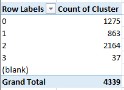


**10**



**K = 5 (5 Clusters)**

Fig. Count of 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

**11**

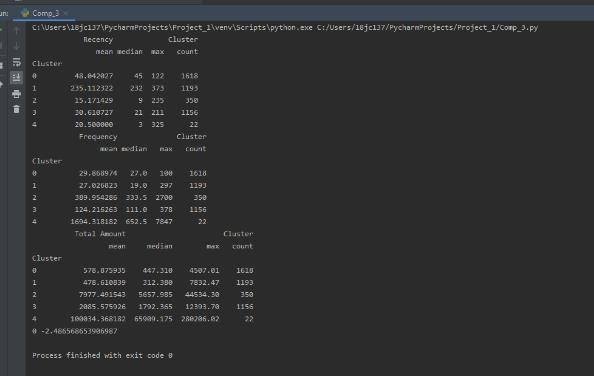


Fig Clusters with their statistical measures

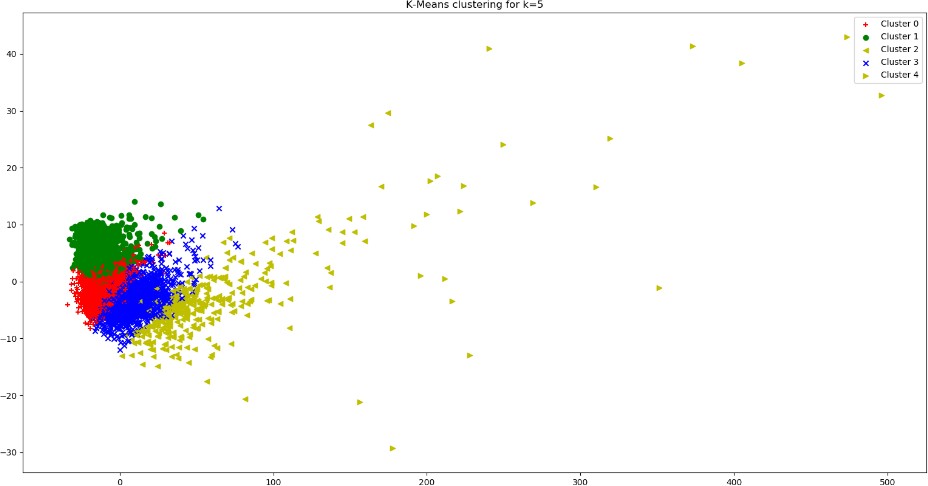
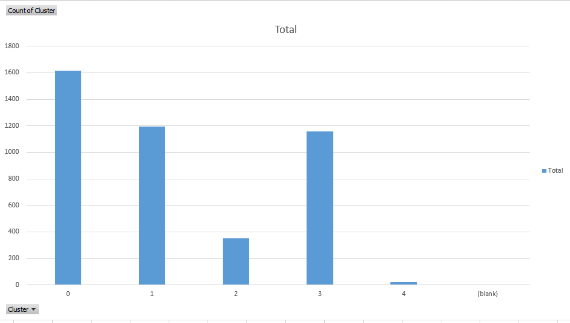


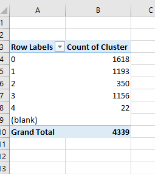
Fig Cluster Plot for K= 5

**12**



**K = 6 (Six Clusters)**

Fig Count of Cluster for K = 5



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

**13**

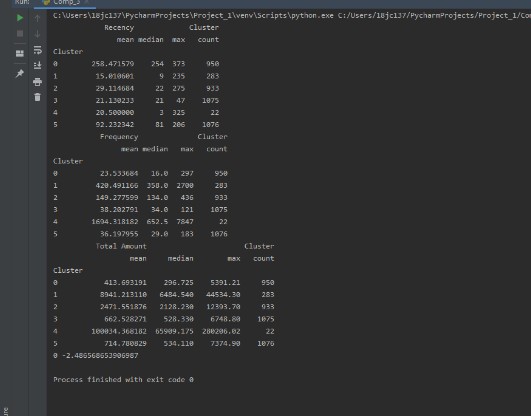
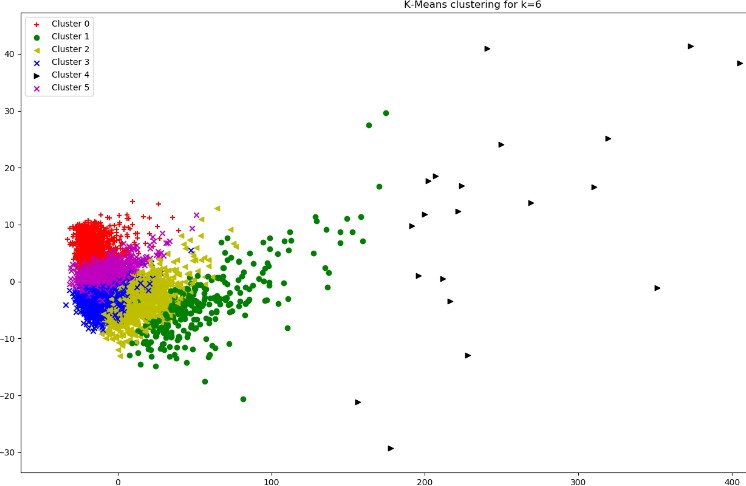
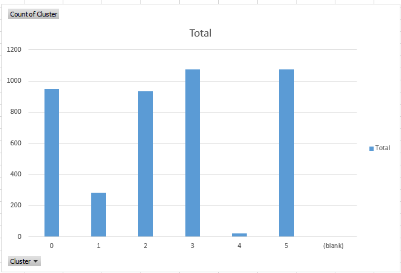


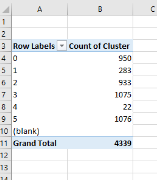
Fig Six Cluster with statistical measures

**14**



**15**



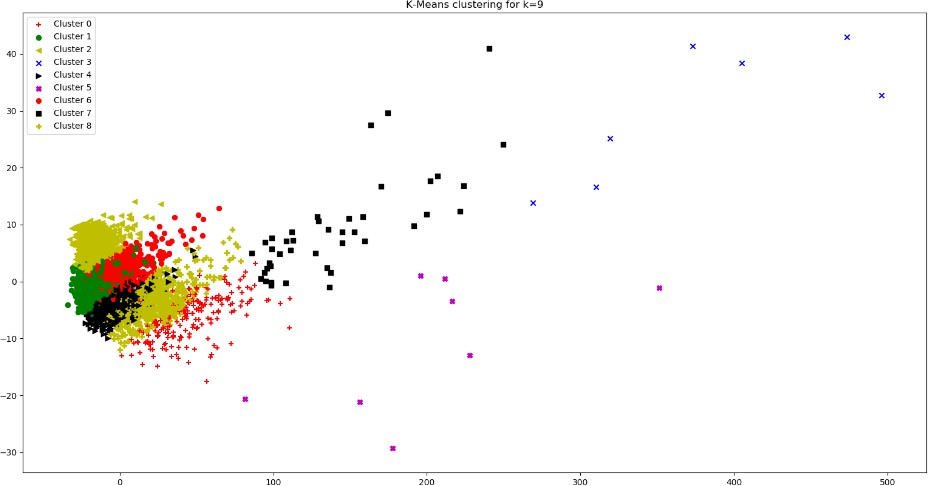


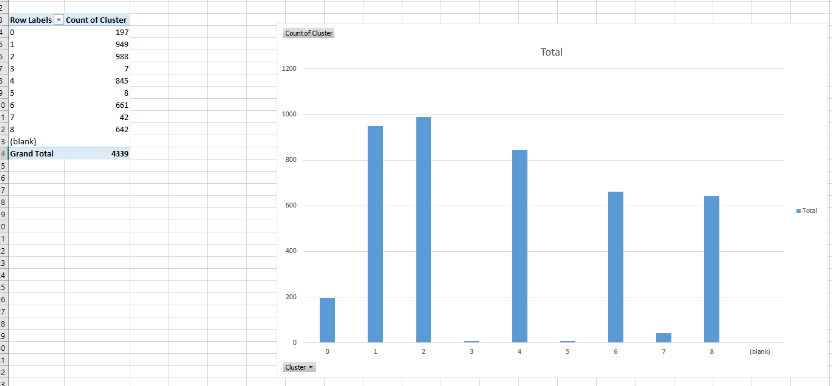
**16**

**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





**17**

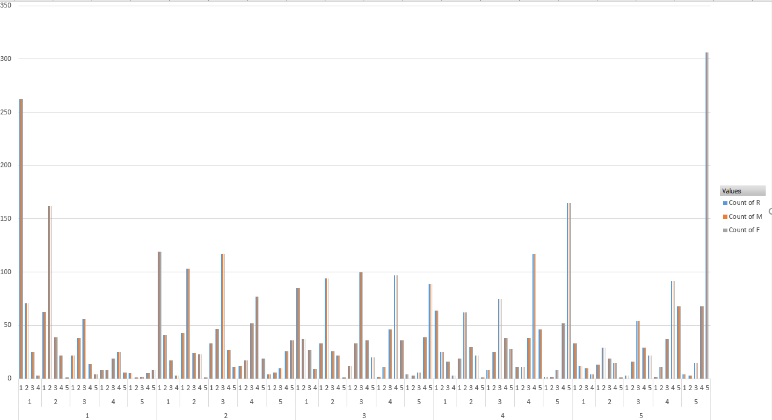


Fig RFM Distribution for all permutations of RFM

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>

**18**