PRML
Minor Project
Online Retail Clustering

Vansh Agarwal B21AI042
Navneet Meena B21CS051

Data Preprocessing & Visualisation

The dataset given in the question has been used for the above problem. The dataset on inspection is found to have large number of null values for CustomerID & Description. **Null entries are handled by dropping them.**
The dataset is grouped on the basis of country. The country wise contribution towards total revenue has been depicted in Fig1, with United Kingdom being the biggest contributor.

Feature Engineering

There are lot of entries having returned items indicated by negative unitPrice or quantity. These are dropped. A new column TotalPrice = UnitPrice * Quantity is introduced. Further, the dataset is grouped on the basis of CustomerID and aggregated on the sum of their TotalAmounts & count of frequencies. Thus TotalAmount & frequency become the most relevant features.

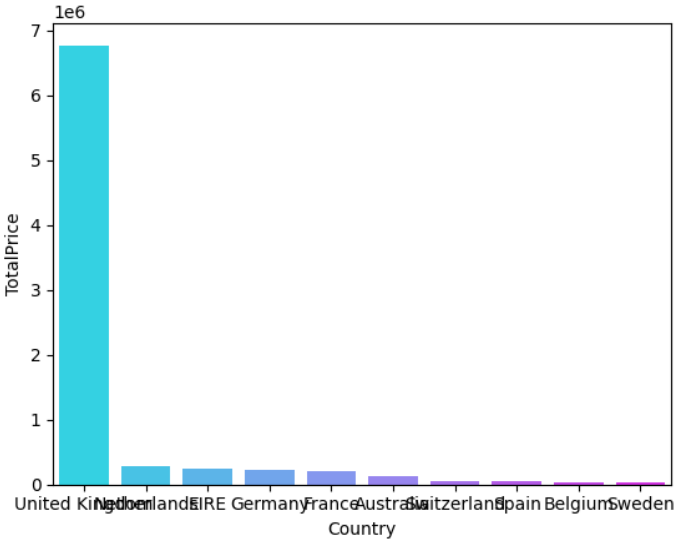

Fig 1.

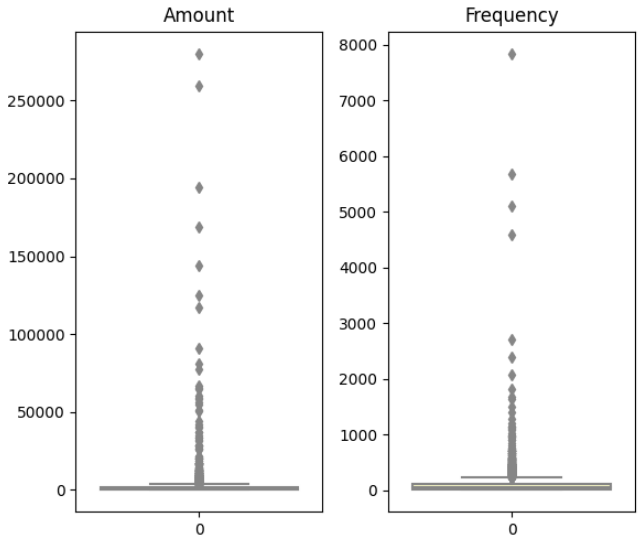

Fig 2

Outlier Detection

Box & Whisker plots have been used to detect the outliers. The upper edge of box shows the upper quartile value (value below which 75% data falls) and the lower edge shows the lower quartile value(value below which 25% data falls). The middle line represents the median value and the singular points show the outliers.
From Fig2 we can infer that the dataset is comprised of large number of outliers. **The total number of outliers is found to be 434.**

Isolation forest have been used to deal with the outliers. Isolation forest is a group of decision trees called isolation trees. The samples that travel deeper into the tree are less likely to be outliers as they required more cuts to isolate them. Similarly, the samples which end up in shorter branches indicate outliers as it was easier for the tree to separate them from other observations.

The anomaly scores are calculated using Isolation forest and the outliers were filtered out from the dataset. The outliers after this processing are indicated in Fig 3.

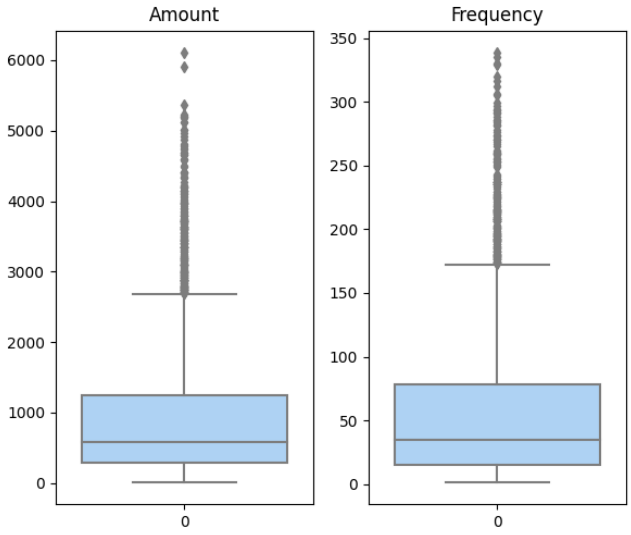

Fig 3

Standard Scaling

The dataset has to be scaled before it is fed into clustering algorithms. This is because **the features having higher ranges will have more influence and fluctuations on the clusters so formed**. This might overshadow some important and more relevant small ranged features. Thus standard scaling has been performed on the dataset.

Optimal Number of Clusters

1. Elbow Method

Elbow method makes a plot of WCSS vs no. of clusters. WCSS (Within Cluster Sum of Squares) is the sum of squared distances of each point in a cluster with their centroid. In order to form better clusters, we need to minimise the WCSS. But on increasing the number of clusters beyond a certain point (elbow) no further significant decrease is observed in WCSS and thus, this is taken to be the most optimal value of number of clusters.

In our case the optimal number of clusters = 5 as shown in Fig 4.

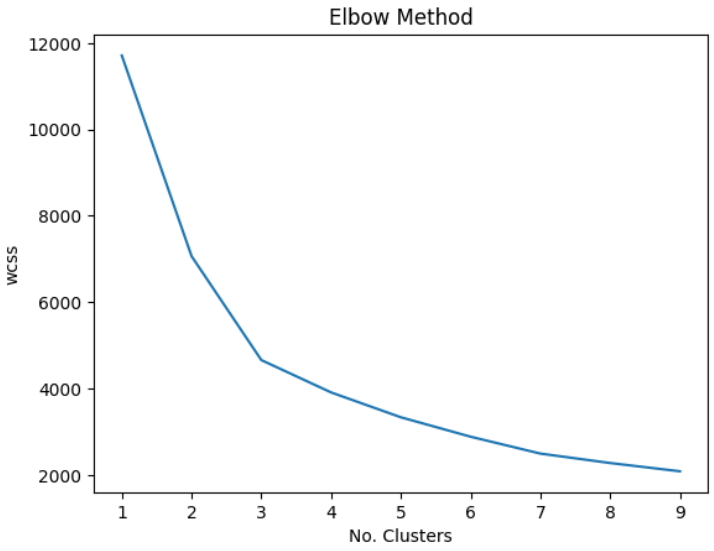

Fig 4

2. Silhouette Analysis
Silhouette Analysis determines how well each object lies in its cluster. Silhouette Score lies in range [-1, 1]. Silhouette score of 1 shows that clusters are very dense and nicely separated. Score 0 means clusters are overlapping. Score < 0 means something is wrong with the data and clusters.

The bands should lie above the dotted line (which shows the average score). The thickness of bands also determines the quality of clustering. In our case the silhouette scores are shown in Fig5. **Although one of the clusters has exceptionally thick bands, the most optimal value for number of clusters is 5 - 6.**

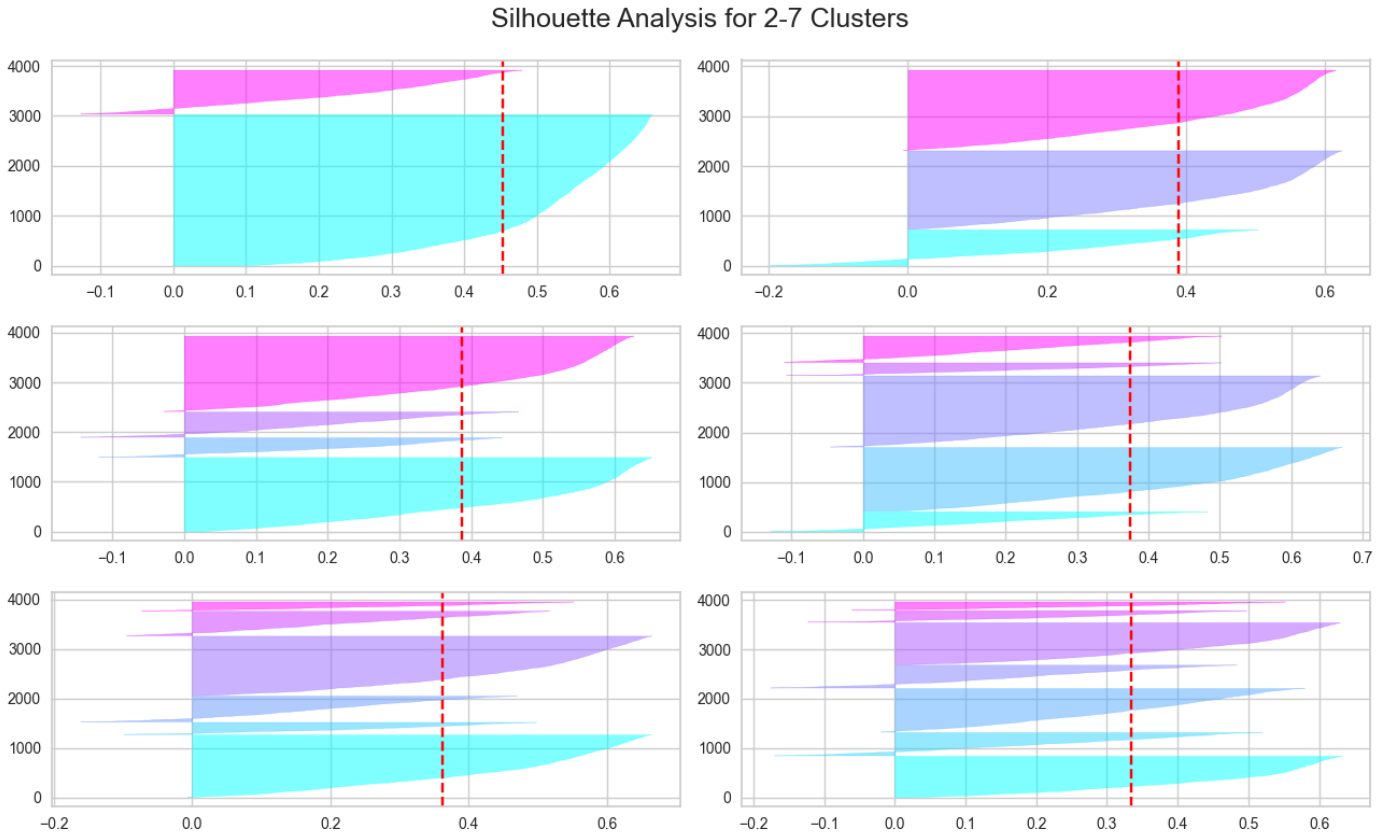

Fig 5

Clustering & Performance Evaluation

The following three clustering algorithms have been implemented whose scatterplots have been depicted in Fig 6, 7 & 8 respectively. For unsupervised learning we don't have any true labels for the clusters. **However we have used Silhouette Scores for determining the quality of clusters. A score closer to 1 indicates better quality of clusters.** The scores for the models are:

- **KMeans - 0.4739**
- **Hierarchical Clustering (Agglomerative) - 0.4255**
- **DBSCAN - 0.278**

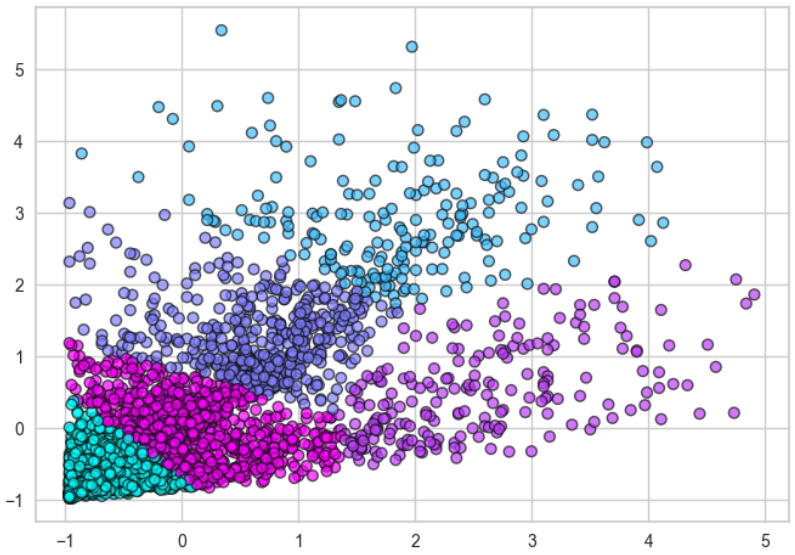

Fig 6 KMeans

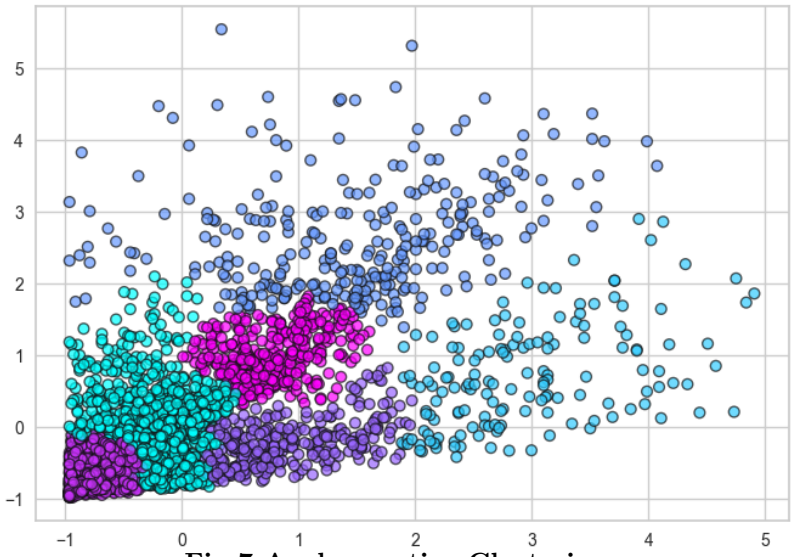

Fig 7 Agglomerative Clustering

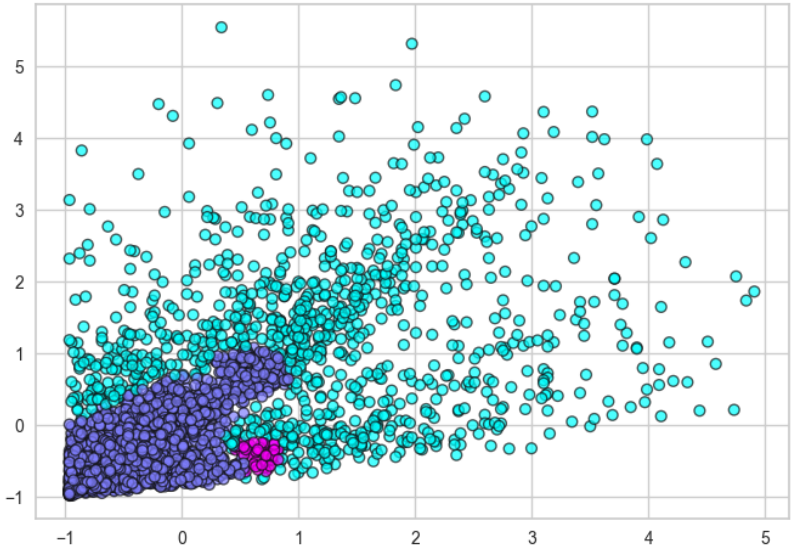

Fig 8 DBSCAN