# Clustering Task 2 Report

## 1.0 Introduction

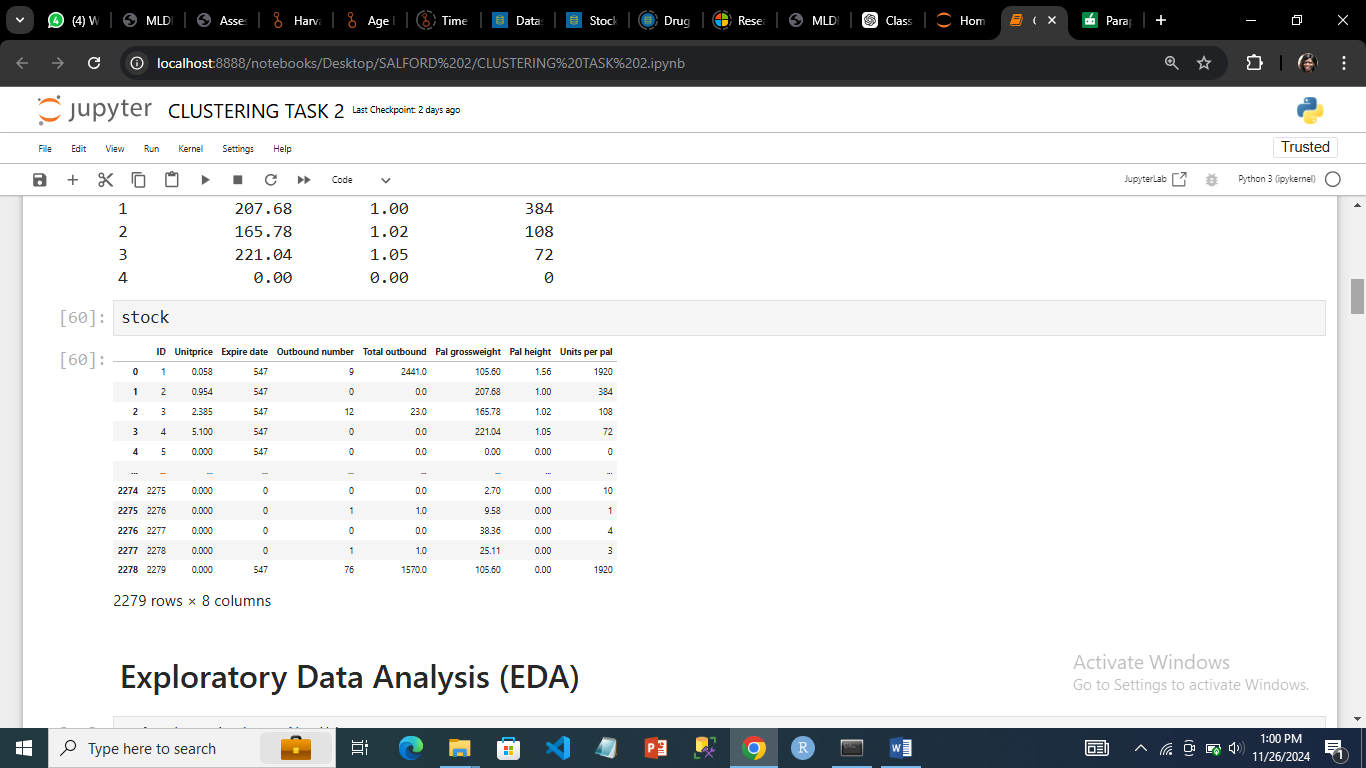
Clustering is an unsupervised machine learning technique used to group data points into clusters based on their similarity (Ester, 1996). This method is particularly effective in exploratory data analysis, where it reveals hidden patterns or groupings within datasets. Unlike classification, clustering does not depend on labeled data, making it useful in a variety of fields such as customer segmentation, anomaly detection, and market analysis (MacQueen 1967).

This report explores the application of clustering algorithms to a dataset, focusing on identifying meaningful clusters and providing insights into the underlying data structure. Using widely adopted clustering techniques such as K-Means and DBSCAN Clustering, the analysis aims to segment the data and uncover valuable patterns (Liao,2005).

2.0 Dataset Description and Exploratory Data Analysis

The dataset was gotten form the UCI repository. It consisted of 2279 observations with 7 features. The features of this dataset is only numerical. The features dated from 2017-02-06 to 2018-02-13. The columns of this dataset include ID, Unitprice, Expire date, Outbound number, Total outbound, Pal grossweight (pallet weight), Pal height (pallet height), Units per pal (number of units per pallet).

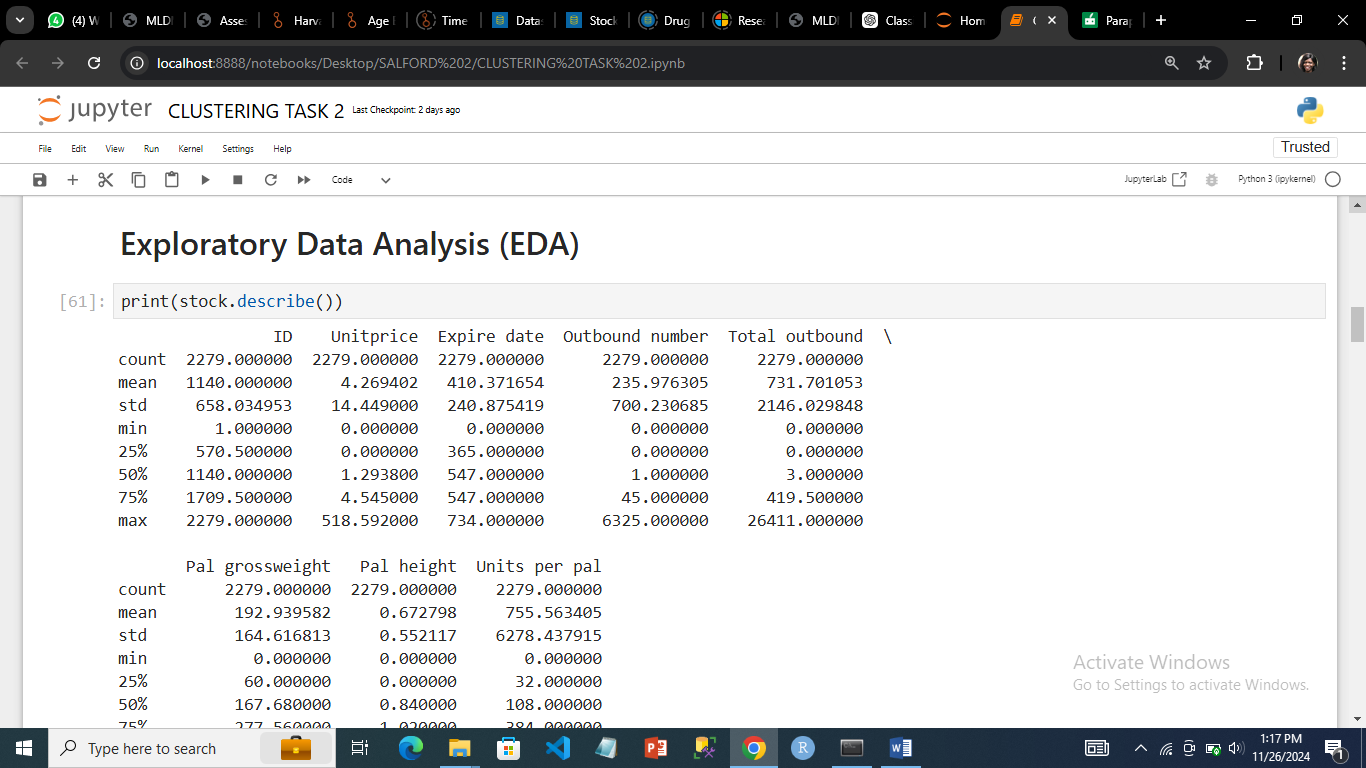
The dataset was analyzed and found that it contains several features representing characteristics of the entities under consideration. Initial exploratory data analysis (EDA) was performed to understand the distribution of features, detect missing values, and identify correlations as this step was critical for ensuring the quality and reliability of subsequent clustering analysis (Liao,2005).



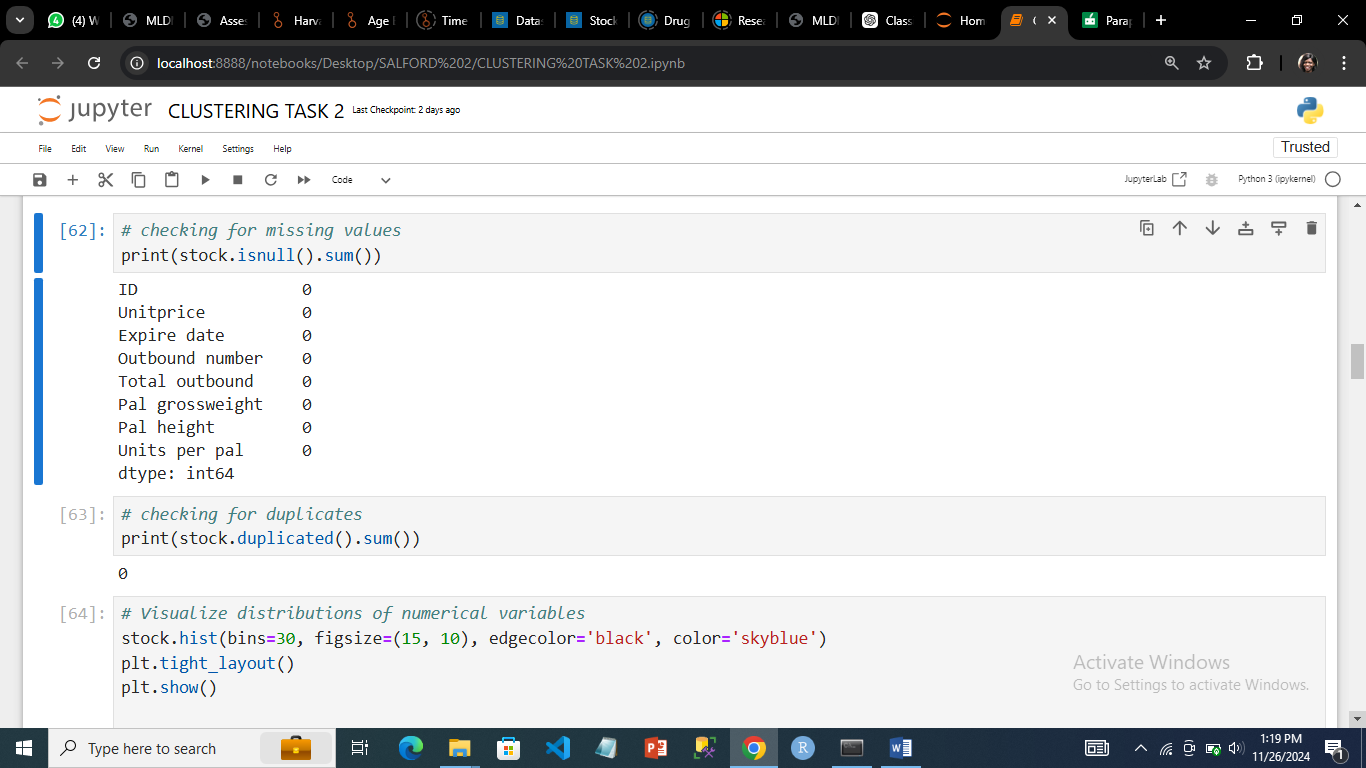
2.1 Exploratory Data Analysis (EDA):

EDA was conducted to understand the dataset's structure and feature distributions. Initial steps involved inspecting the dataset for missing values, detecting outliers, and visualizing relationships between features. Features were plotted to assess their range and variability, ensuring a comprehensive understanding before clustering (Kaufman, 2005).

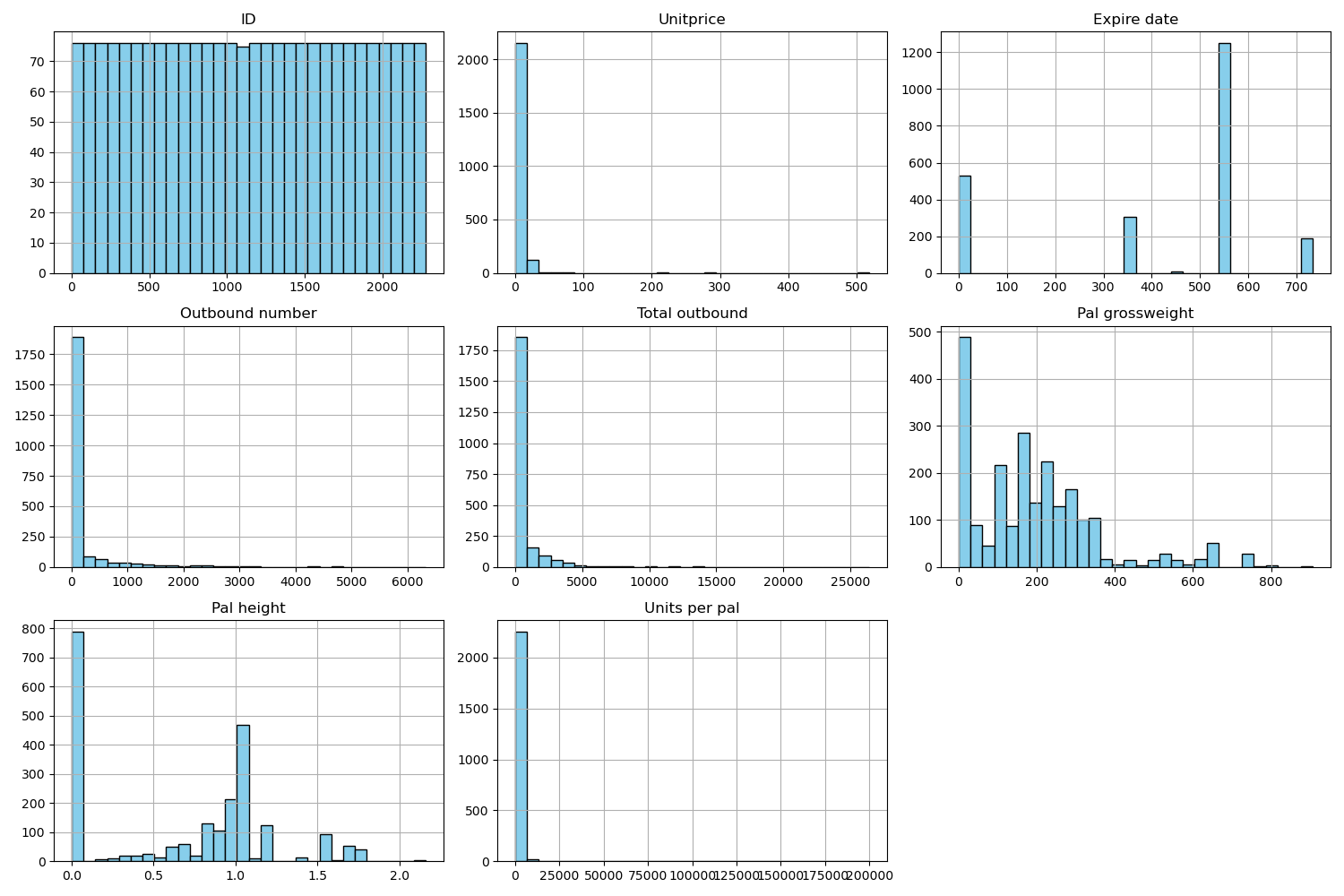
The dataset contained no missing values. Feature correlations were examined to identify redundancies, ensuring that highly correlated features did not skew the clustering results.



***Describing the data***

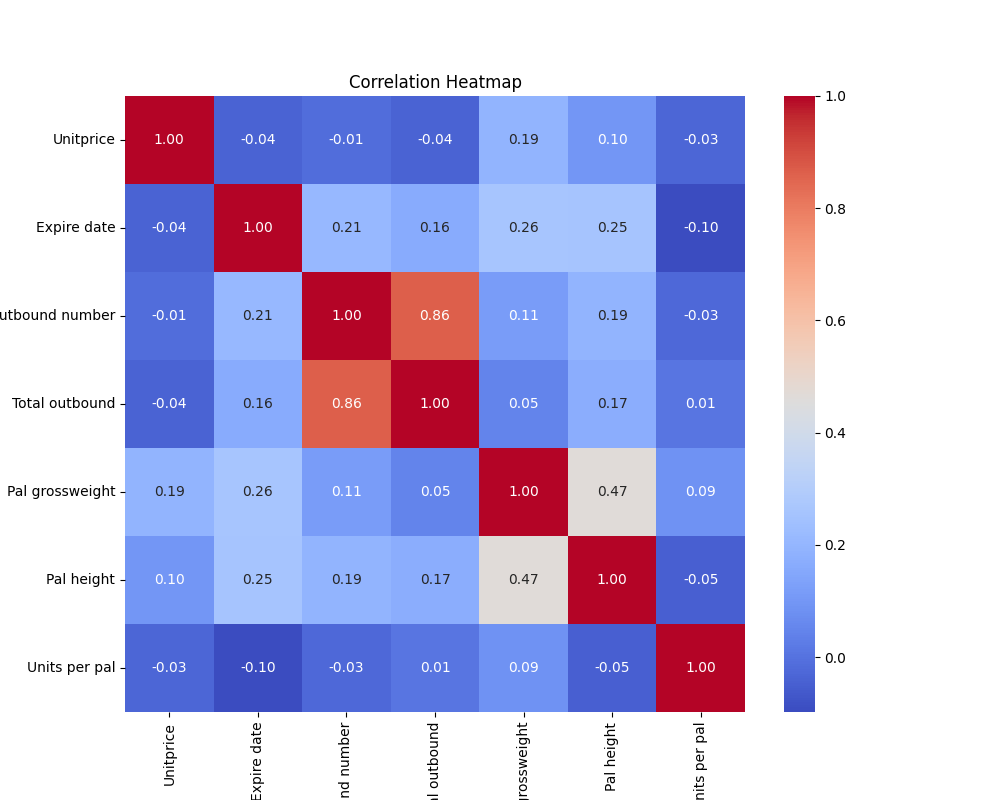


***Checking for missing values and duplicates***



***Distribution of the dataset***

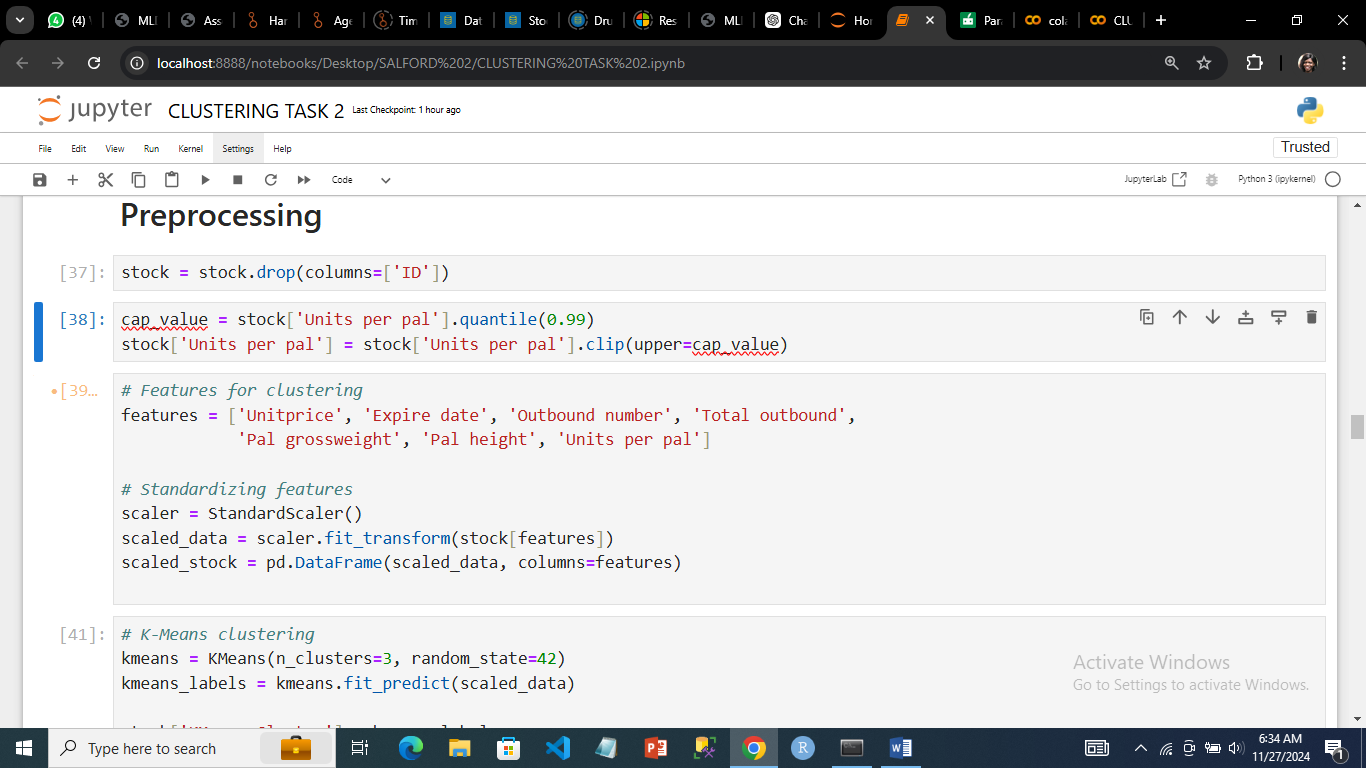
The heatmap illustrated how strongly the features in the dataset are related to one another. While "Units per pal" and "Expire date" exhibited nearly no connection (light blue), and showed independence, "Outbound number" and "Total outbound" had a significant correlation (dark red), which suggested a strong linear relationship.

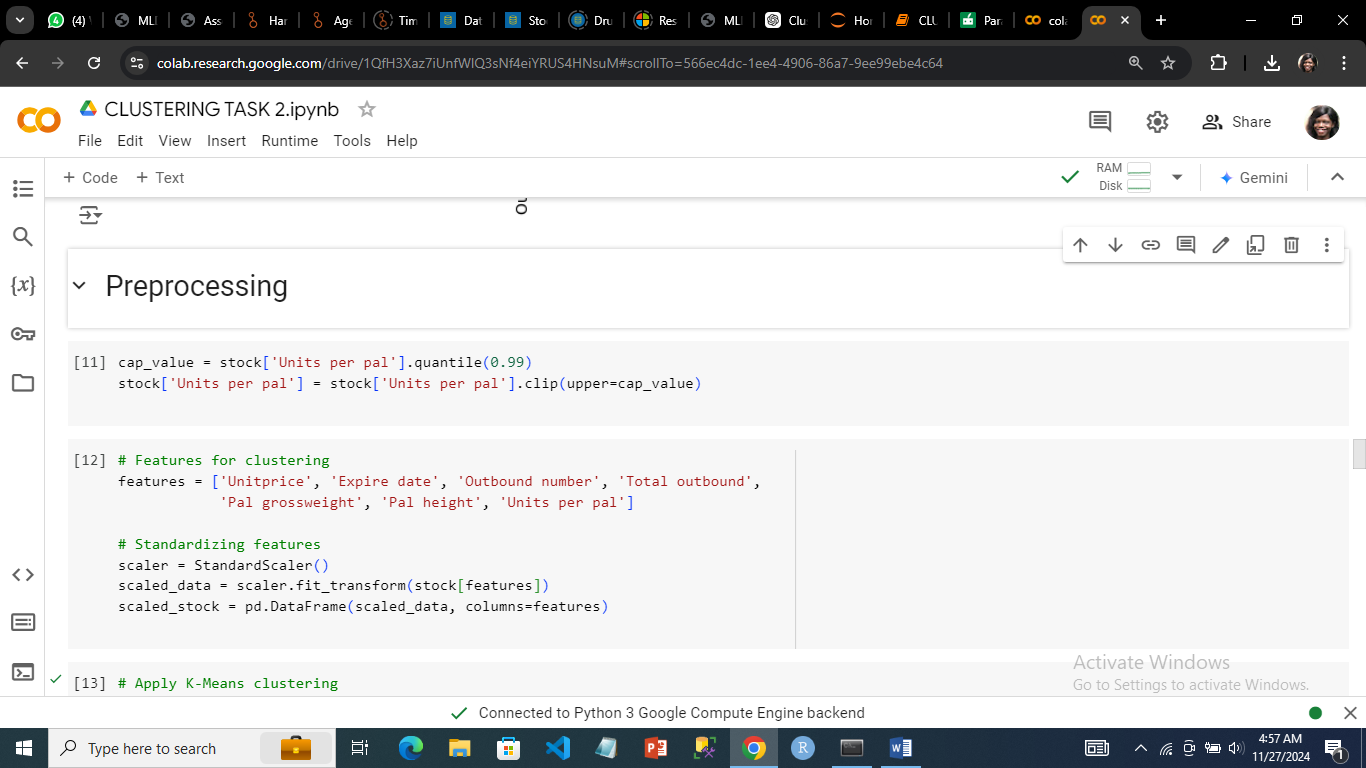


***Heatmap showing correlation***

## 3.0 Data Preprocessing

Data preprocessing involved cleaning the dataset and preparing it for clustering. For this analysis, outliers were removed, and numerical features were standardized using StandardScaler to ensure that each variable are on a similar scale. Standardization was vital for this analysis because of the distance-based clustering algorithms like K-Means that we employed and this is to prevent features with larger scales from dominating the clustering process (Kaufman, 2005). For this analysis, K-mean and **Density-Based Spatial Clustering of Applications with Noise(DBSCAN) was employed.** DBSCAN was used because it is a popular clustering algorithm in machine learning and data mining. It groups together points that are close to each other based on a measure of density and identifies points that are outliers (noise) (Kaufman, 2005).





***Reduction of outliers and standardization***

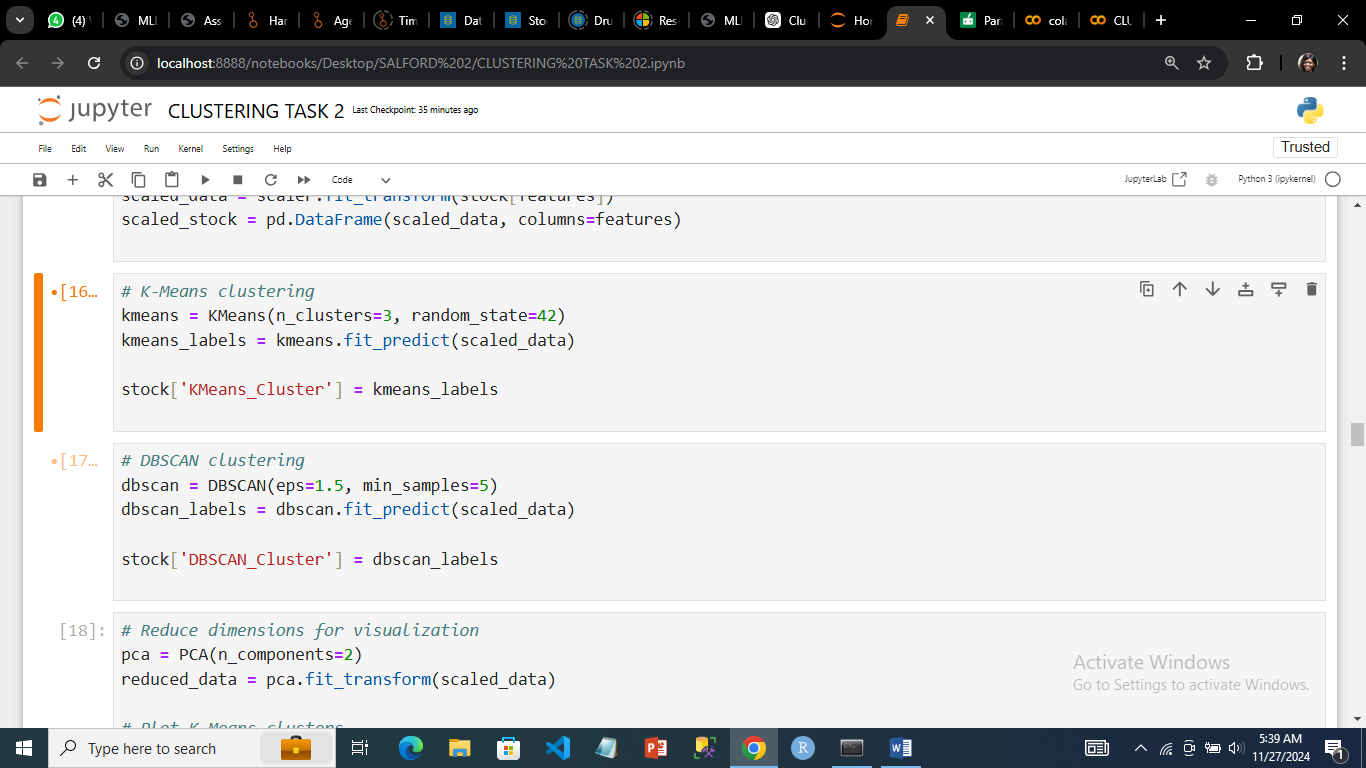
## 4.0 Clustering Algorithms

## 4.1 K-Means Clustering

The data was segmented into various categories using K-Means clustering. To ensure consistency of findings, the algorithm was initialized with three clusters and a fixed 'random\_state' value of 42. The clustering method entailed fitting the scaled data to the K-Means algorithm and forecasting the cluster labels for each data point.

The generated cluster labels were allocated to a new column in the dataset called 'KMeans\_Cluster' to link each data point to its related cluster. This technique classified the dataset into three separate clusters using unsupervised learning, allowing for deeper examination of data patterns and linkages.

The reason why was used was because partitions the dataset into a predefined number of clusters. It works by iteratively assigning data points to the nearest cluster centroid and updating the centroids to minimize the variance within clusters. The Elbow Method was employed to determine the optimal number of clusters by evaluating the within-cluster sum of squares (WCSS) for different values of K.



***K-means clustering***

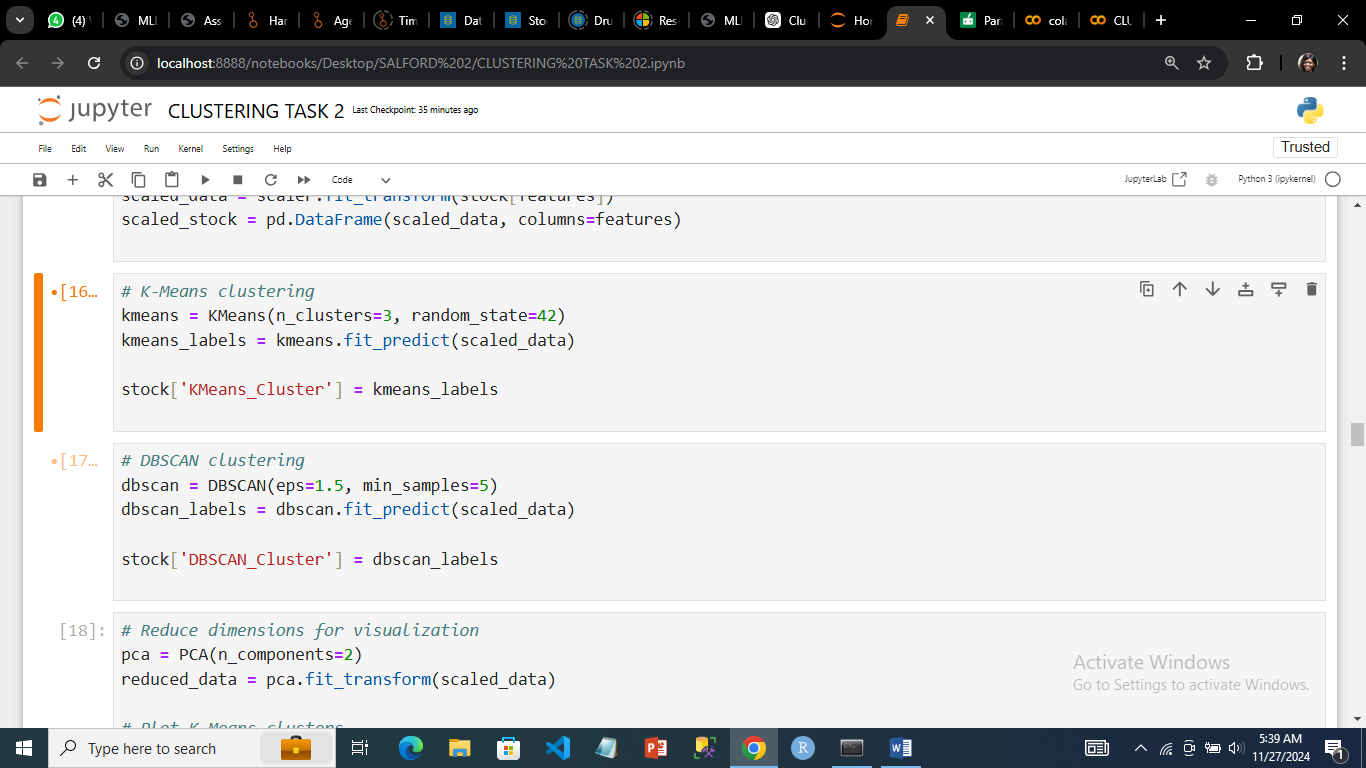
4.2 Density-Based Clustering (DBSCAN)

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) identifies clusters in data based on density. Unlike traditional methods, DBSCAN does not assume clusters to have specific shapes and can detect arbitrarily shaped clusters and outliers. This algorithm uses two key parameters: eps, which defines the maximum distance for points to be considered neighbors, and min\_samples, which specifies the minimum number of points needed to form a dense cluster.

DBSCAN was applied in this analysis to group data into clusters. Points within dense regions were categorized into clusters, while sparse regions were classified as noise, making DBSCAN particularly effective for handling outliers. The resulting clusters reflect underlying density variations in the data without requiring a pre-specified number of clusters, allowing the algorithm to adapt naturally to the dataset's structure.

DBSCAN was applied to identify clusters in the dataset based on density. The algorithm was configured with an eps value of 1.5, which defines the maximum distance for points to be considered part of the same neighborhood, and a min\_samples value of 5, specifying the minimum number of points required to form a dense region.

The algorithm was fit to the scaled data, and cluster labels were assigned to each data point, including a label of -1 for outliers or noise points that did not belong to any cluster. These cluster labels were stored in a new column, DBSCAN\_Cluster, within the dataset. This analysis provided insight into the data's structure, capturing clusters of varying shapes and densities while identifying outliers effectively.



***Density based clustering (DBSCAN)***

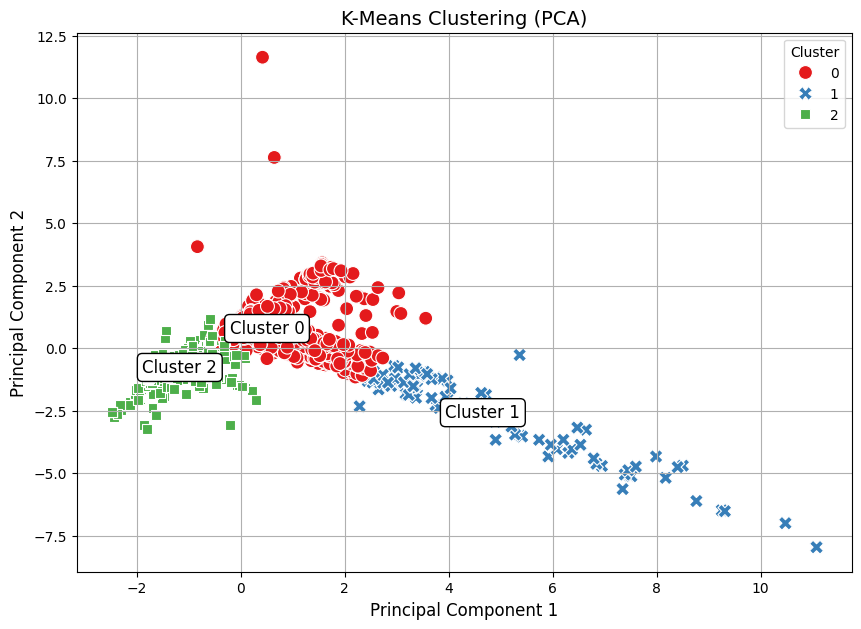
4.3 Visualization of K-means and DBSCAN

In order to visualize the clusters created by the K-Means and DBSCAN algorithms in a two-dimensional space, dimensionality reduction was carried out in the study using Principal Component study (PCA). The high-dimensional data was reduced to two principle components by applying the PCA with two components (Kaufman, 2005).

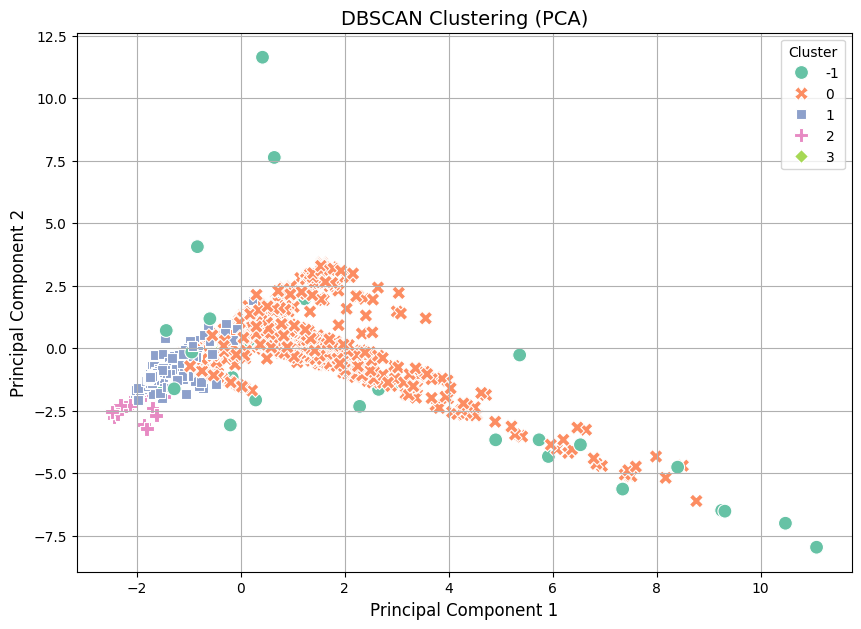
The K-means clusters was produced. Each data point in the scatter plot was colored according to the cluster label that the K-Means algorithm had assigned it. Using various colors on the plot, the visualization made it evident how the K-Means algorithm had divided the data into discrete clusters.

The results of the DBSCAN clustering were displayed in the second plot. A scatter plot was utilized, with dots colored according on their DBSCAN cluster labels, much like the K-Means plot. Because DBSCAN is a density-based clustering method, it may have found clusters with different densities and forms. It also classified outliers as a distinct category, which was shown visually.

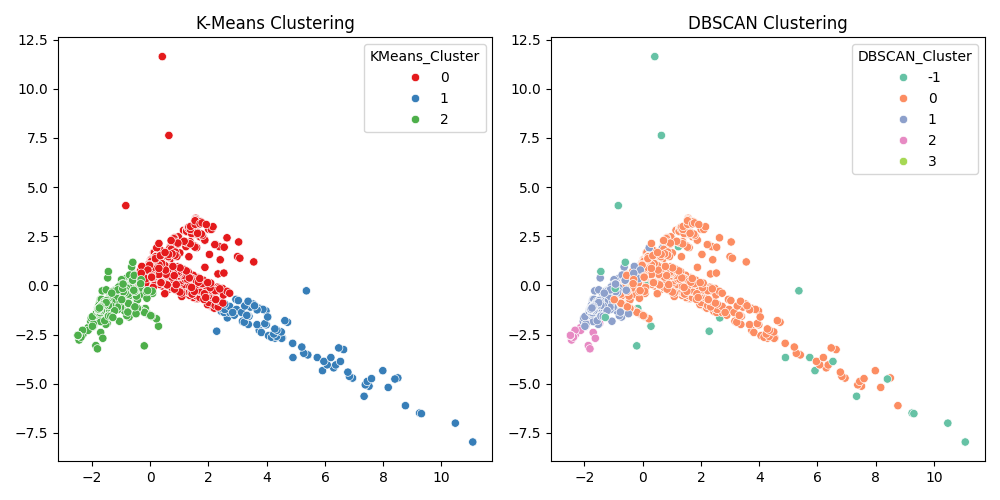
A comparison of the K-Means and DBSCAN algorithms' performance in clustering the data was shown by the two graphs. After making precise layout tweaks to maximize space and presentation, the figure was shown. The plots were arranged such that they were exhibited side by side for simpler comparison.

******

***K-means clustering***



***DBSCAN clustering***



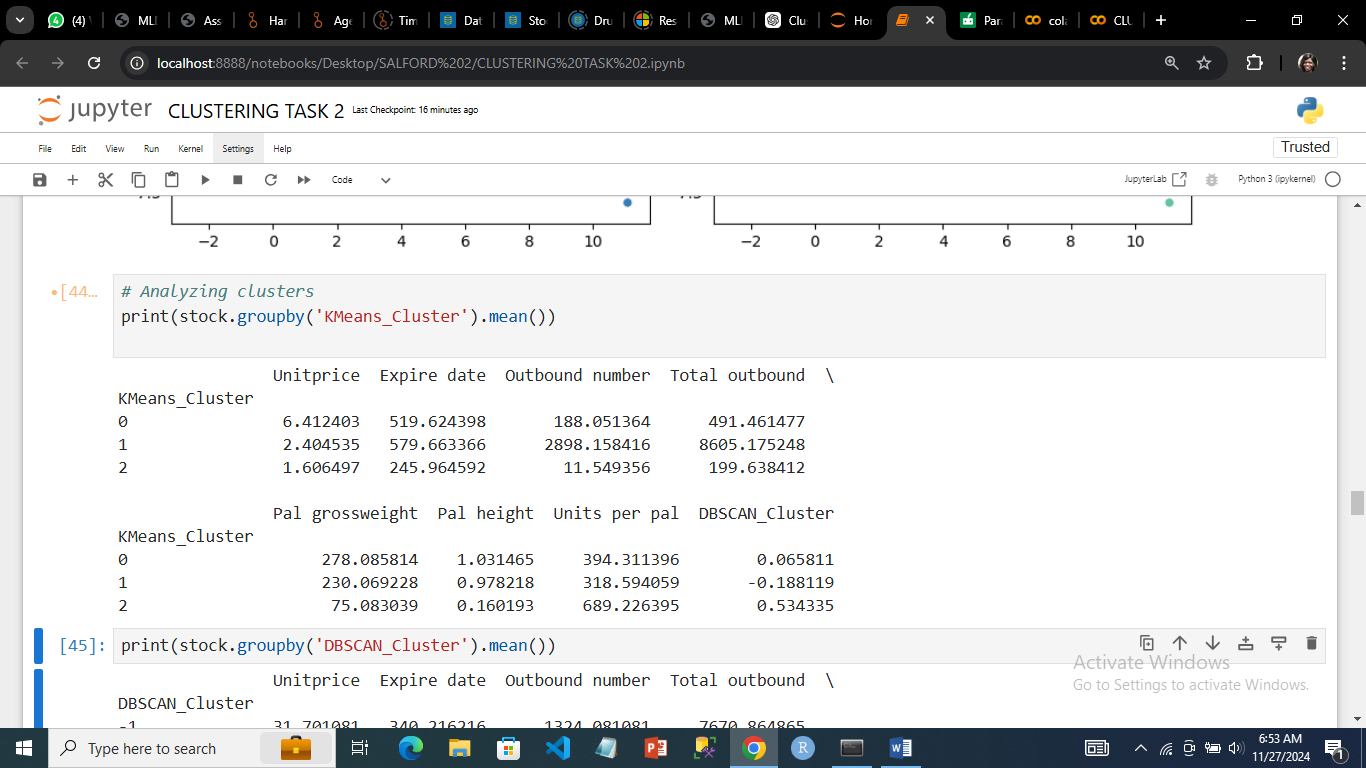
***K-means and DBSCAN clustering***

4.3 Analysis of K-means and DBSCAN

#### KMEANS

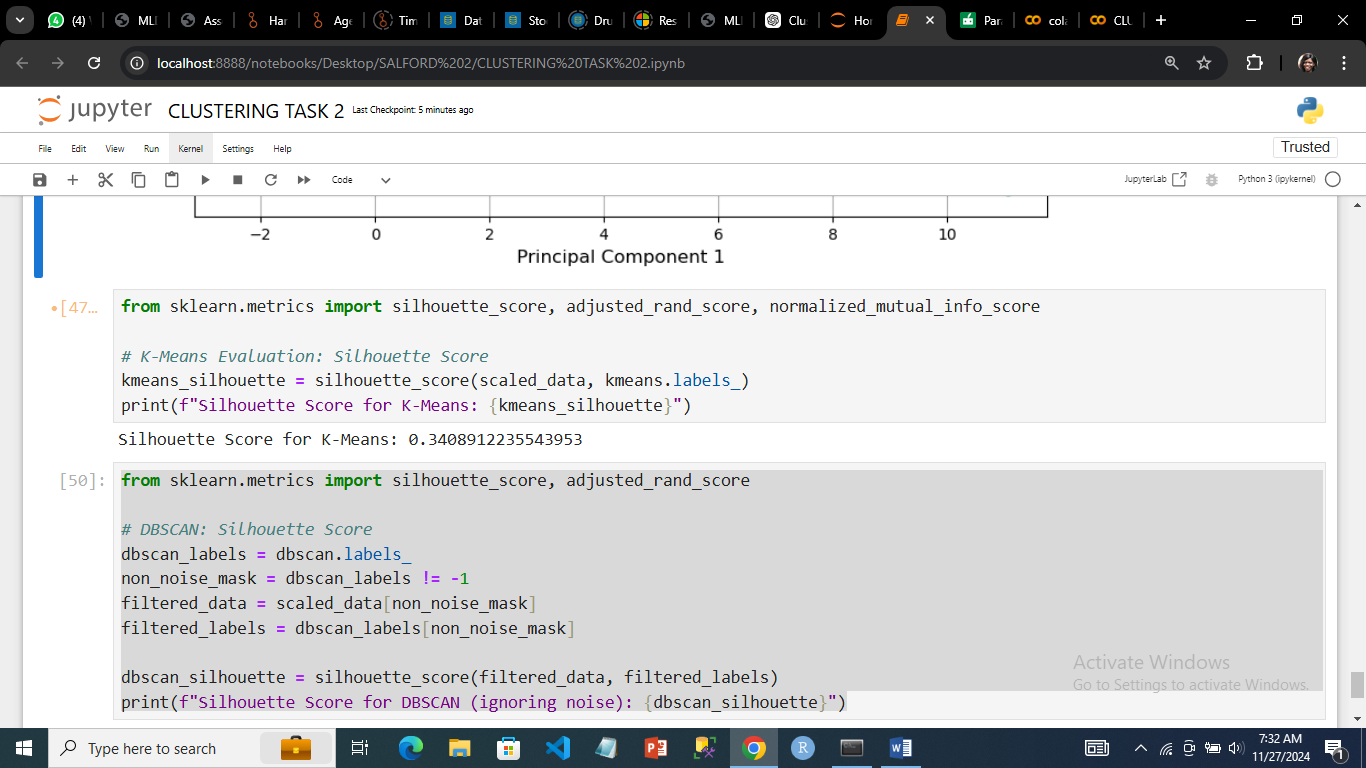
In order to display the average values of different attributes for every cluster, the cluster centroids for the K-Means and DBSCAN clustering methods have to be examined. Unit price, expiry date, outbound number, total outbound, pallet gross weight, pallet height, and units per pallet were among the attributes whose mean values were represented by the centroids for K-Means across the data points in each cluster. In comparison, the values for these traits were lower in cluster 2, higher in cluster 1, and generally intermediate in cluster 0.

The DBSCAN cluster values, on the other hand, demonstrated the density-based character of the algorithm; certain points were categorized as outliers or noise (shown by negative cluster labels), while others were divided into distinct clusters.



***K-means cluster mean***

The degree of distinction between the groups is indicated by the K-Means clustering silhouette score, which is 0.3409. The score denoted a moderate level of clustering quality.

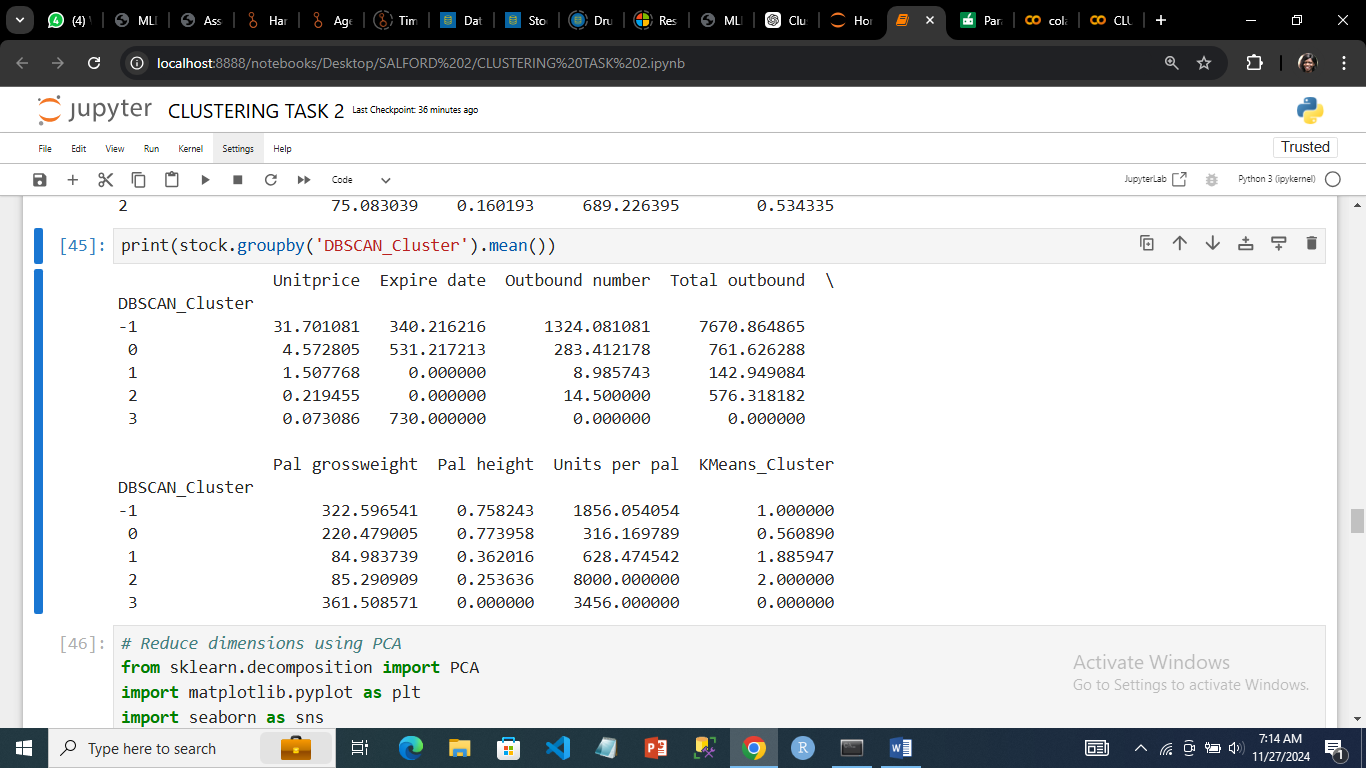


***Silhouette score for K-means***

#### DBSCAN

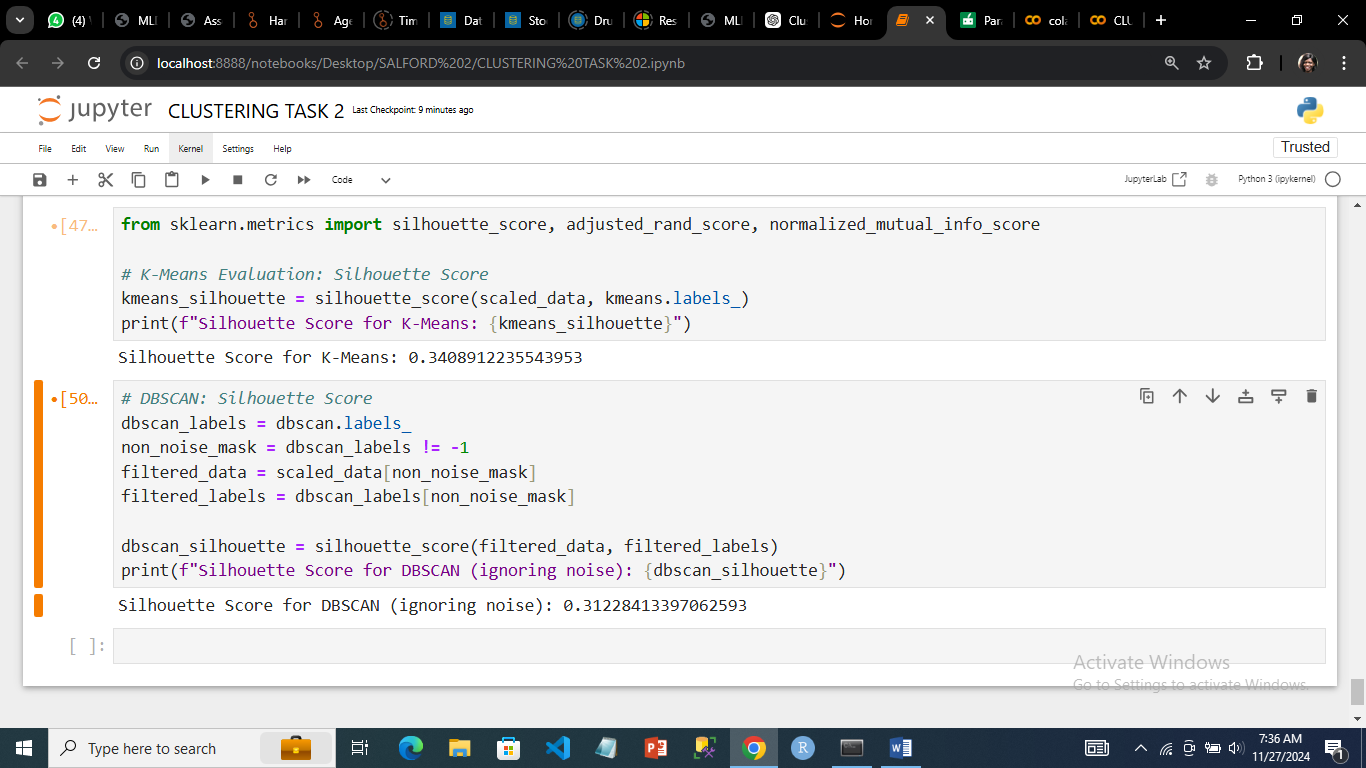
The DBSCAN and K-Means methods were used to determine the centroids of different characteristics for every cluster (Liao,2005). With much higher values for unit price and total outgoing, cluster -1 stood out from the other clusters in the DBSCAN clustering results, indicating that it contained noise or outliers. While cluster 3 had a zero value for pallet height and units per pallet, its high value for pallet gross weight set it apart from the other DBSCAN clusters (0, 1, 2, and 3) in terms of attributes like outbound numbers and units per pallet.

In contrast, the K-Means clusters showed distinct patterns: cluster 2 had a much lower unit price and more units per pallet, cluster 1 showed greater unit prices and total outbound, and cluster 0 was characterized by reasonably moderate values across characteristics. The data points were categorized differently by the centroids of DBSCAN and K-Means. While DBSCAN concentrated on dense regions, finding outliers and changing cluster sizes based on density, K-Means probably separated the data into more evenly distributed clusters, resulting in more irregular groups with some solitary points (cluster -1). The comparison demonstrates how the two algorithms' basic presumptions regarding cluster forms and density affect how they interpret and arrange the data.



***DBSCAN cluster mean***

DBSCAN's silhouette score, after eliminating noise points, is 0.3123. This suggests intermediate cluster separation, implying that while the clusters are separate, there is still some overlap or potential for improvement.



***Silhouette score for DBSCAN***

## 5.0 Conclusion

Clustering analysis gave useful insights into the dataset's structure, revealing different categories. The findings of this study revealed the efficiency of clustering approaches in identifying hidden patterns from data, paving the path for targeted interventions or optimization (Liao,2005).

In this analysis, K-Means and DBSCAN clustering algorithms were applied to the stock dataset from the UCI repository containing various features such as "Unitprice," "Expire date," and "Total outbound." amongst others. Both algorithms were assessed using the Silhouette Score to evaluate their clustering quality.

**K-Means Clustering:** The K-Means algorithm was set with three clusters. The Silhouette Score of 0.3409 suggests that the clusters formed by K-Means were distinguishable but still exhibited a level of overlap. This indicates that while K-Means was able to partition the data into clusters, the separation between them can be improved (Kaufman, 2005)

**DBSCAN Clustering:** The DBSCAN algorithm, which is density-based, was applied with the parameters eps=1.5 and min\_samples=5. The Silhouette Score of 0.3123, after excluding noise, indicated that the clustering also showed moderate separation between clusters. However, DBSCAN's ability to identify noise points, represented by the label -1, and its use of density to define clusters offer advantages in identifying arbitrary-shaped clusters. This flexibility allowed DBSCAN to potentially handle more complex cluster structures compared to K-Means.

## References

1. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 226-231.
2. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1(14), 281-297.
3. Liao, T. W. (2005). Clustering of time series data—a survey. Pattern Recognition, 38(11), 1857-1874.
4. Kaufman, L., & Rousseeuw, P. J. (2005). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley-Interscience.
5. Sklearn Documentation: DBSCAN and KMeans Clustering. Retrieved from <https://scikit-learn.org/stable/modules/clustering.html>