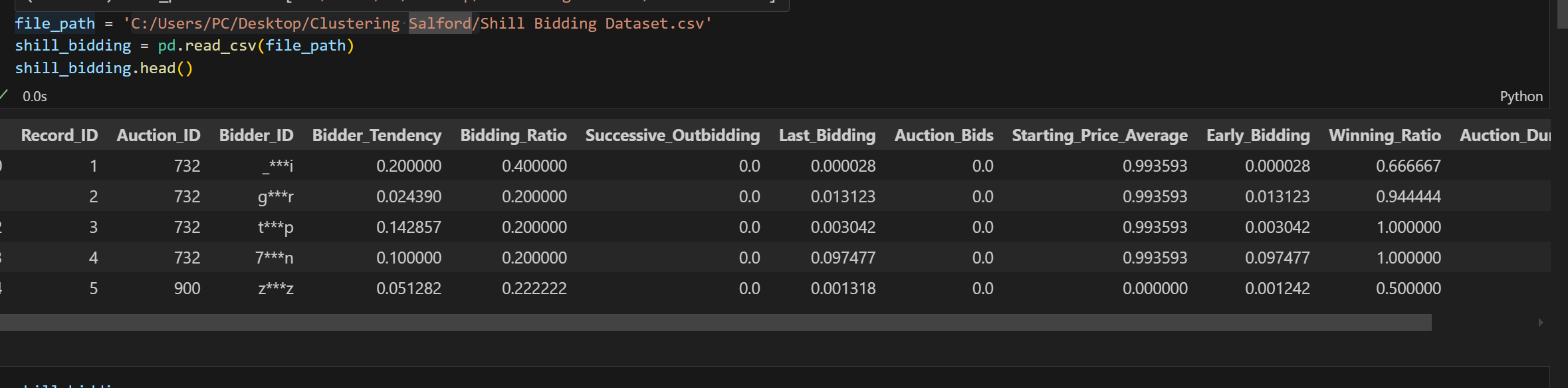
# Clustering Analysis of Shill Bidding Dataset Using K-Means and DBSCAN

## 1. Introduction

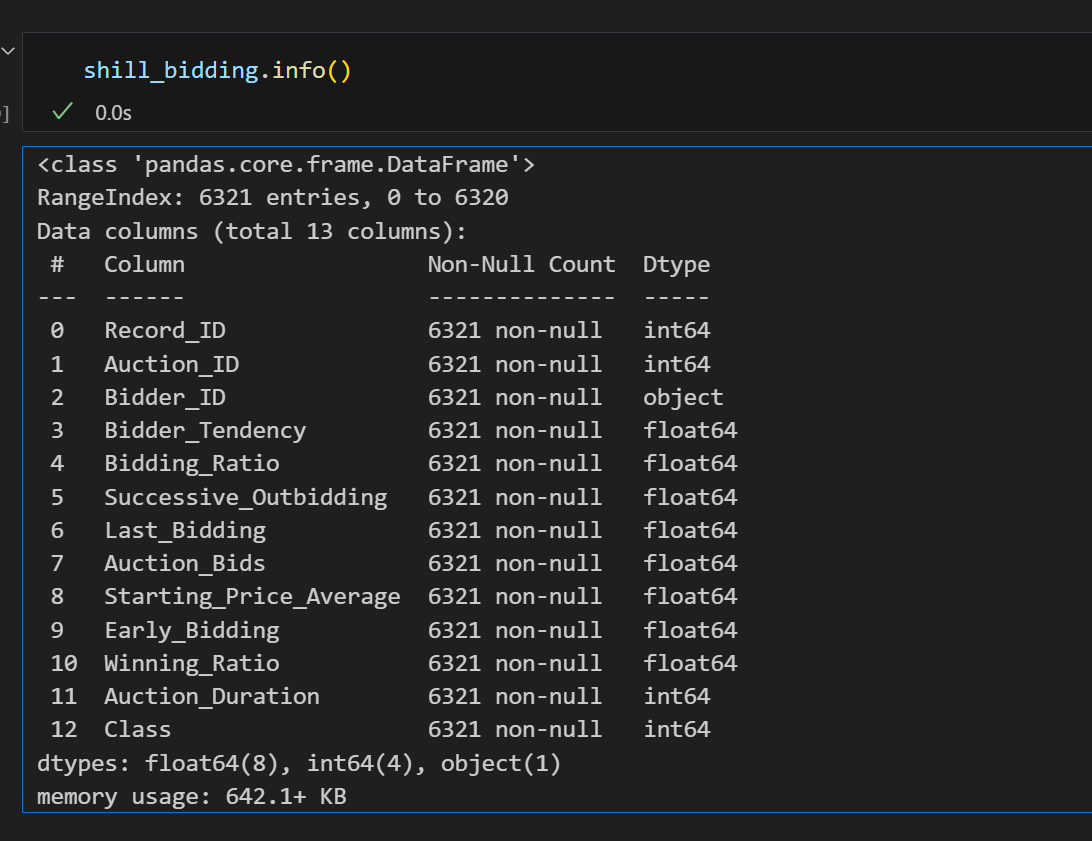
Clustering is an unsupervised learning technique that helps organise data points into separate clusters based on their similarity (Kriegel et al, 2011). The Shill Bidding Dataset was analysed using clustering algorithms to discover underlying trends. The dataset, provided in CSV format, comprised different features linked to bidding activities; however, the bidder's identity was removed for privacy concerns and since it was not required for the study (Han et al, 2012). The clustering study was performed using two different clustering algorithms: K-Means and DBSCAN.

## 2. Dataset Overview

The dataset, Shill Bidding Dataset, was imported into a pandas DataFrame for further analysis. The dataset contains many numeric features; however, the field bidder\_id was removed because it was unnecessary to the investigation. The initial examination of the dataset revealed important details about the structure, such as the number of records and columns. To ensure that the variables were understood well, the dataset was evaluated using the info() method, which provided information about the data types, and .head(), which examined the dataset's first few rows.



*Dataset overview*



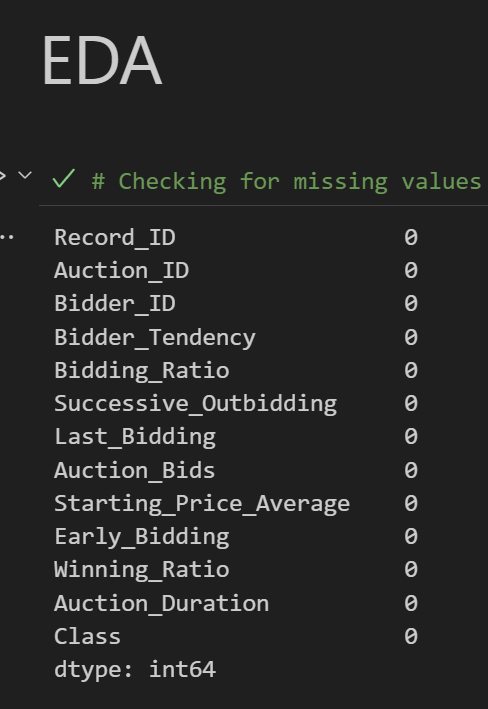
*Dataset info*

## 3. Exploratory Data Analysis (EDA)

To better understand the dataset, the analysis began with Exploratory Data Analysis (EDA). This includes looking for missing values, visualizing the distribution of numeric features, and determining the association between these features (Zhang et al, 2014).

### 3.1 Missing Values

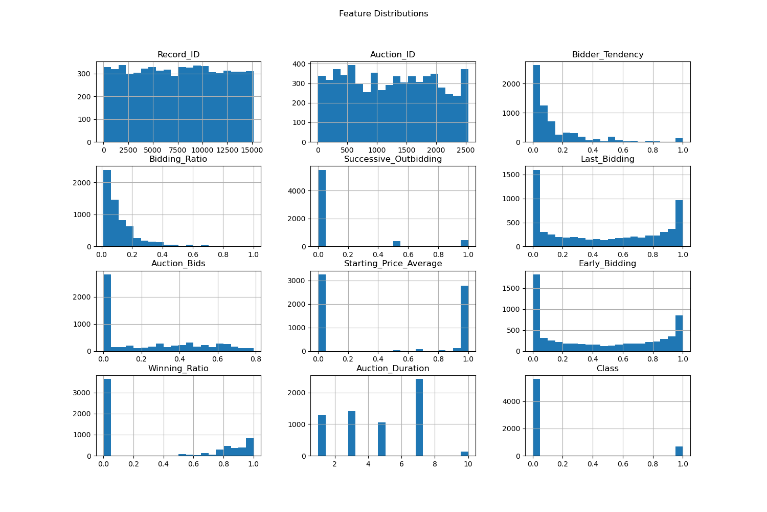
The presence of missing values was determined using the.isnull().sum() function. This step was critical in ensuring that the dataset was clean and suitable for future analysis. There was no missing data (Xu et al, 2005)



*No missing values*

### 3.2 Histograms of Numeric Features

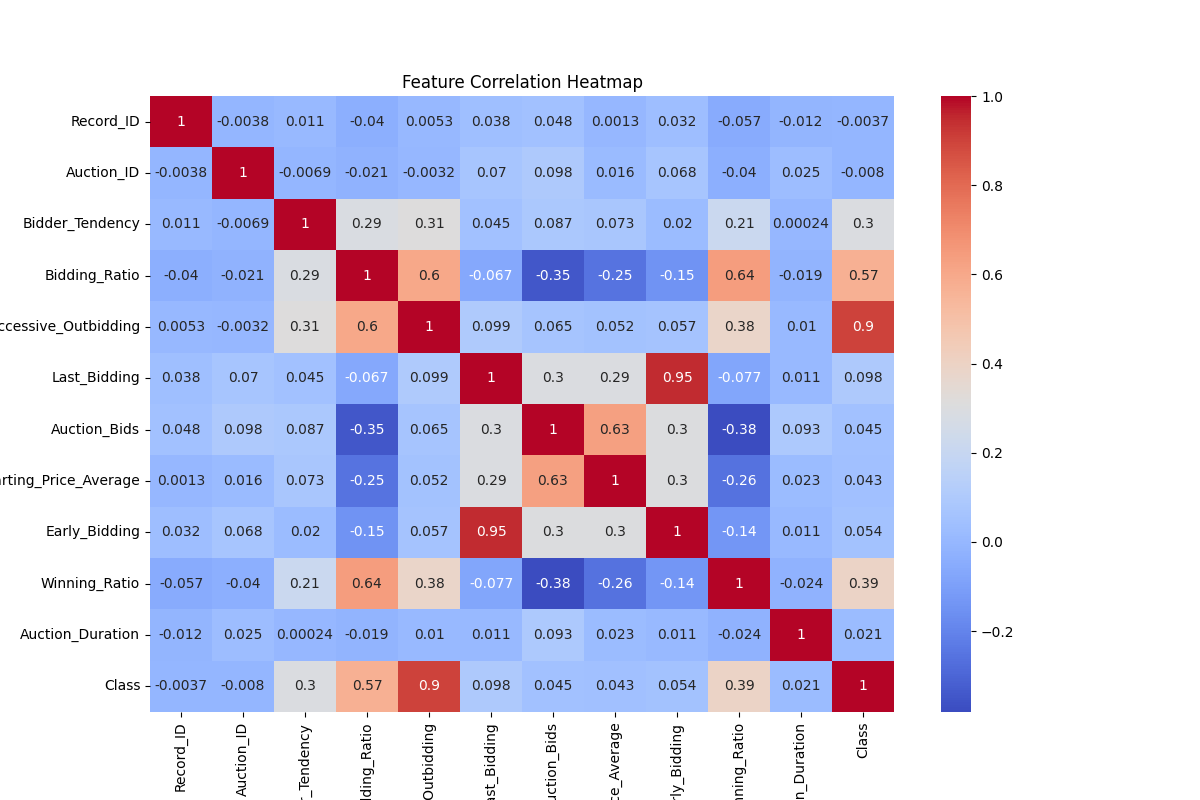
Histograms were created to represent the dataset's numerical features (Gusmão et al, 2015). This was done to better understand the distribution of each attribute and look for skewness, outliers, or trends in the data. (Ester et al, 1996)



*Histogram on data distribution*

### 3.3 Correlation Heatmap

A correlation heatmap was constructed to determine the links between the dataset's various elements (Jain, 2010). The Pearson correlation coefficients between all feature pairs were visualised using seaborn's sns.heatmap() function. The heatmap gave a quick and straightforward approach to identify whether features were strongly connected and might potentially be grouped or used for dimensionality reduction. (Berkhin, 2006)



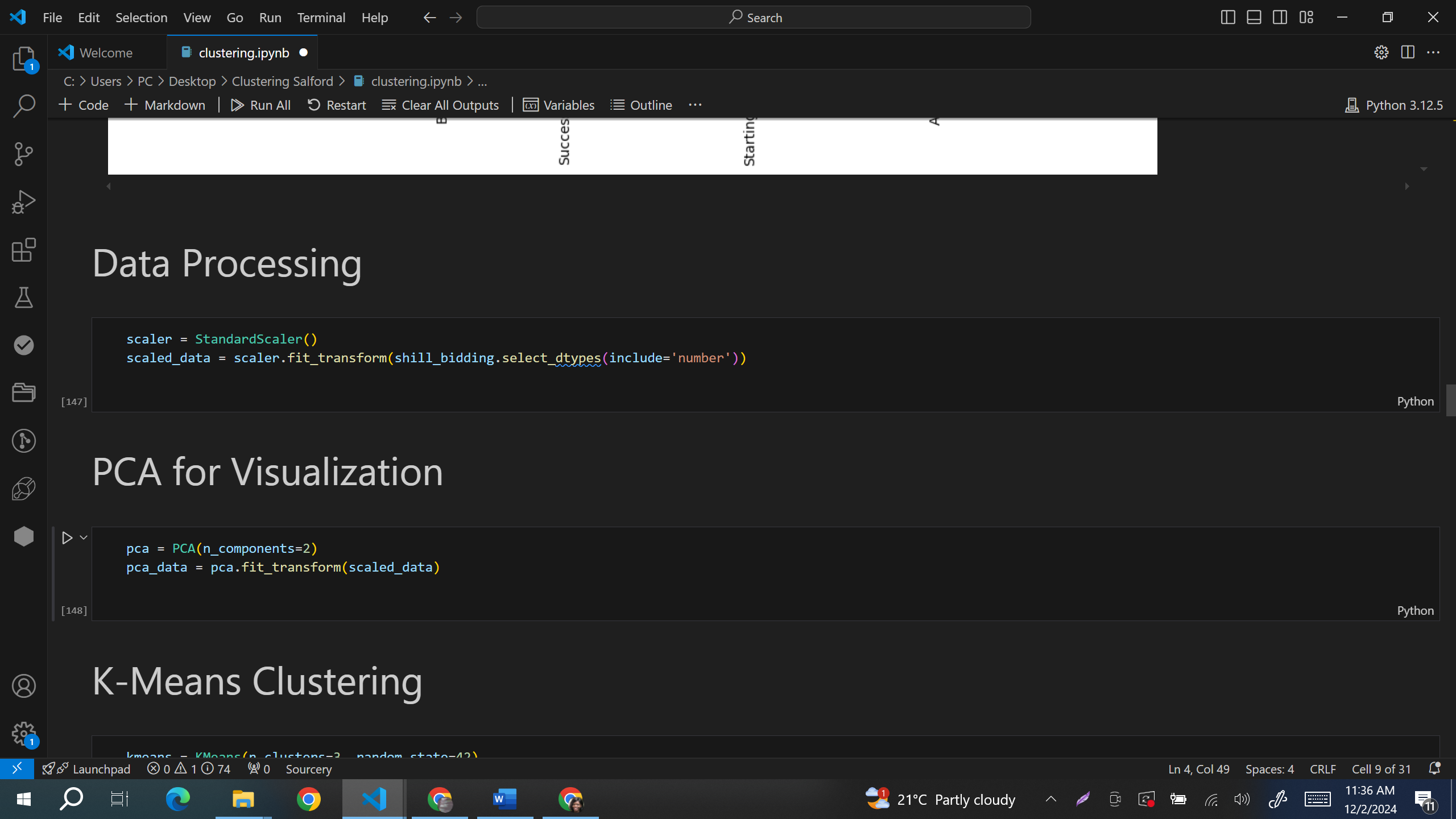
*correlation on data distribution*

## 4. Data Preprocessing

The data had to be preprocessed (Kriegel et al, 2011). This involved scaling the numeric data so that all features were on a similar scale, ensuring that no characteristic dominated the clustering process (Zhang et al, 2014)

### 4.1 Standard Scaling

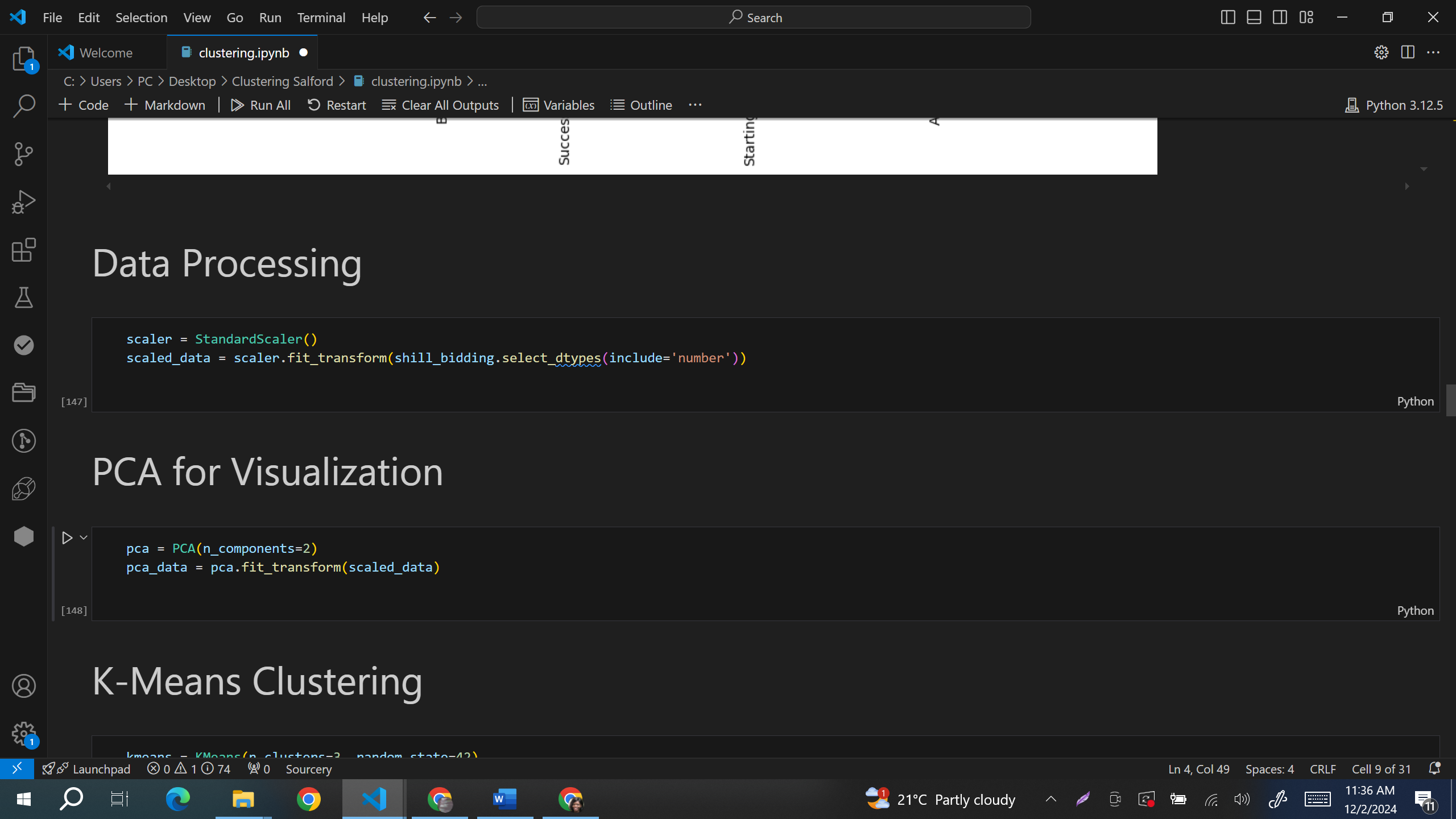
The data was standardized using scikit-learn's StandardScaler, which removed the mean and scaled to unit variance. This is a frequent preprocessing step in machine learning, particularly when utilizing distance-based methods such as K-Means and DBSCAN, to guarantee that all characteristics contribute equally to the clustering process (Ester et al, 1996).The fit\_transform() technique was applied to the dataset's numeric columns, resulting in the scaled data utilized for clustering (Han et al, 2012)



*Standard scaling*

## 5. Principal Component Analysis (PCA)

Dimensionality reduction was carried out using Principal Component Analysis (PCA), a technique that minimizes the number of features while maintaining as much variance as possible (Gusmão et al, 2015). This phase was critical in visualizing the clustering results in two dimensions (Berkhin, 2006). The PCA model was trained on the scaled data, resulting in a feature space to only two principle components (Xu et al, 2005) These two components, which accounted for the most volatility in the data, were utilized to visusalize the clusters in 2D space. The reduced data from PCA was saved as pca\_data for later visualization and clustering. (Jain, 2010)

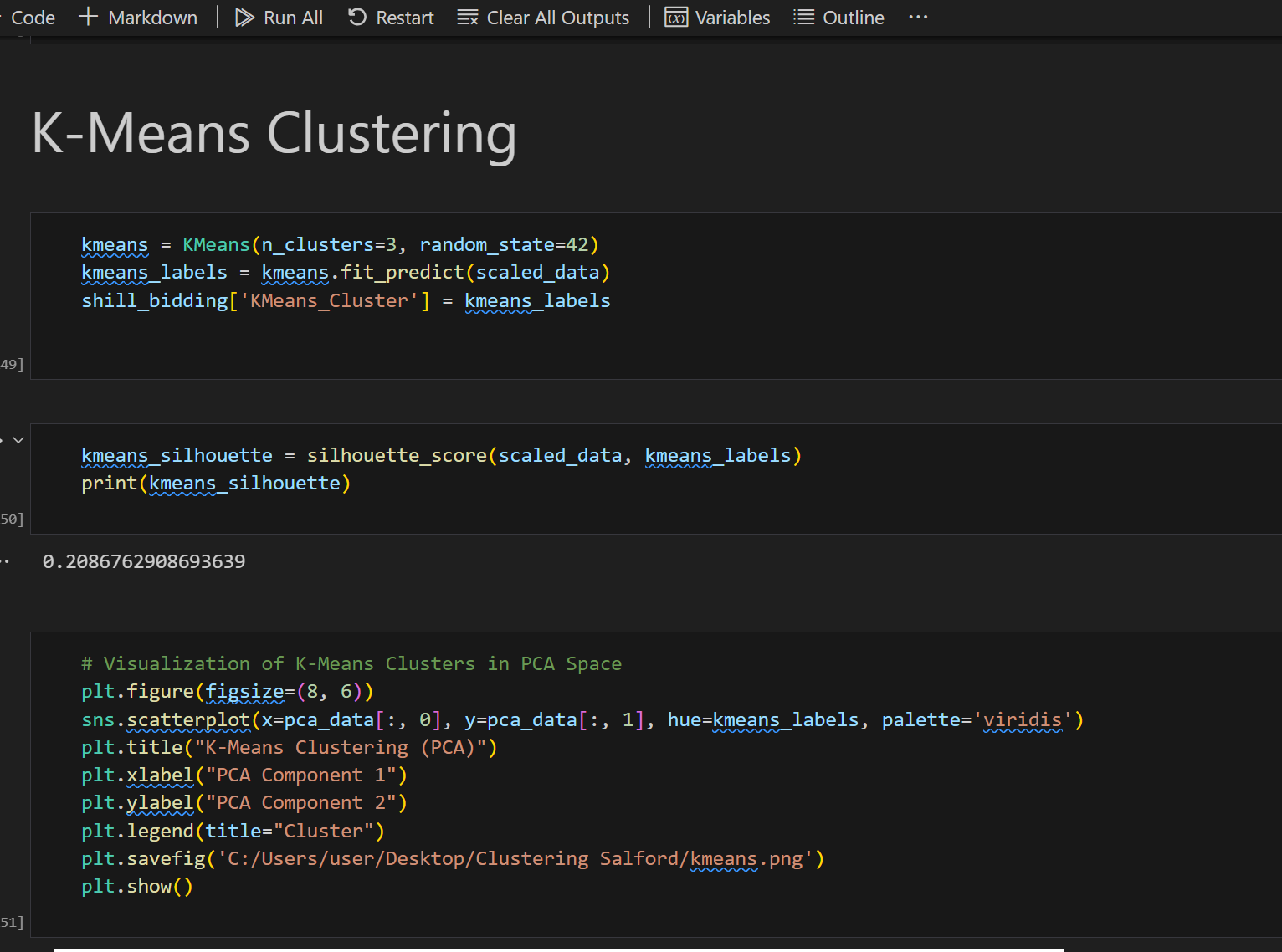


*Principal component analysis*

## 6. K-Means Clustering

The dataset was processed using the K-Means clustering algorithm, with the purpose of categorising comparable bidding behaviours into groups (Han et al, 2012). K-Means is a centroid-based clustering technique that divides data into k clusters while minimising variation within each cluster.

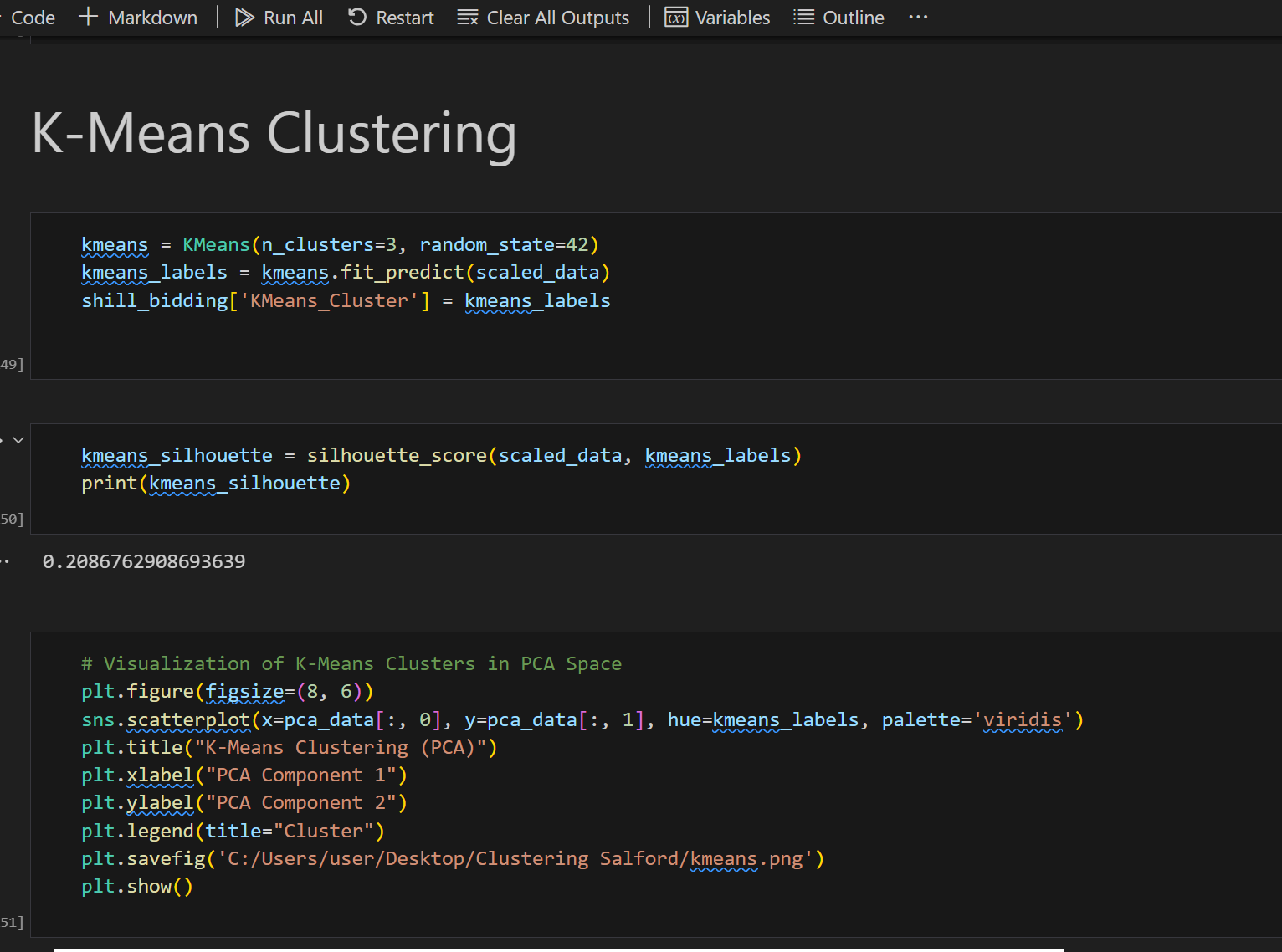
Scikit-learn's KMeans model was employed, with a cluster size of three (Xu et al, 2005). The fit\_predict() method was used to fit the model and provide cluster labels for each data point. These labels were then added to the dataset as a new column named KMeans\_Cluster.



*K-means Clustering*

### 6.1 Silhouette Score

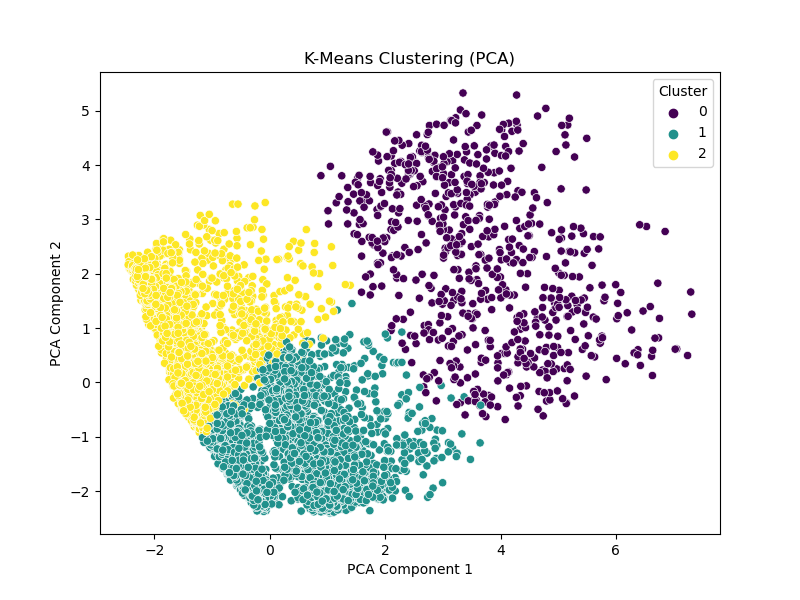
To evaluate the quality of the clustering, the **Silhouette Score** was calculated(Kriegel et al, 2011) . This metric provides an indication of how well-separated the clusters are. The silhouette score of 0.208 in the K-means clustering result indicated that the data points were not well separated between clusters. It showed that some points were likely overlapping between clusters or that the clusters themselves were not strongly separated.



*Silhouette Score for K-MEANS*

### 6.2 K-Means Cluster Visualization

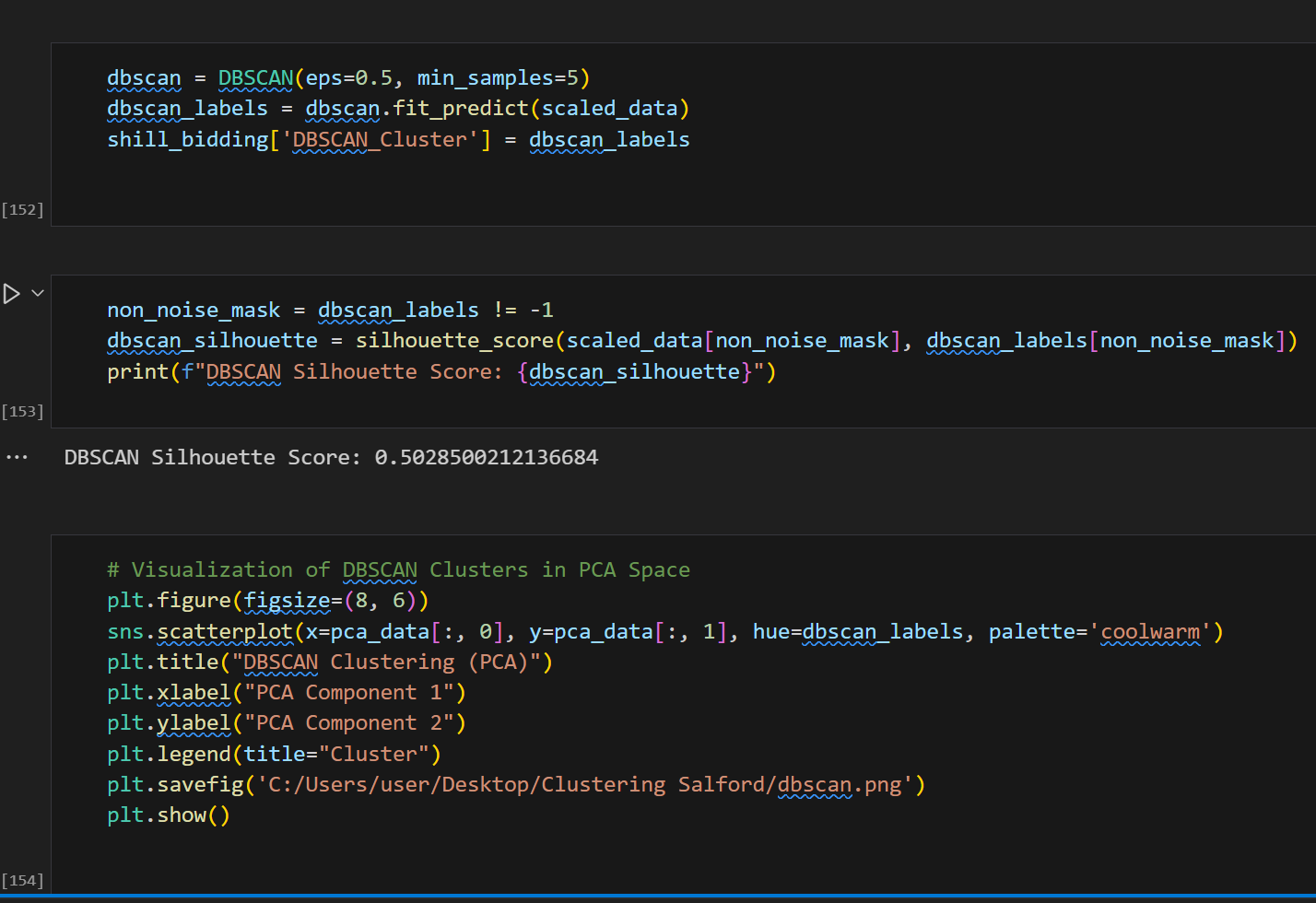
The K-Means clustering results were visualized in the 2D PCA space with seaborn's scatterplot() function. Each data point was color-coded based on its cluster label, with different hues representing the clusters. The graphic clearly demonstrated how the data points were divided into three different clusters in the smaller space. (Zhang et al, 2014)



*K-MEANS Cluster Visualization*

## 7. DBSCAN Clustering

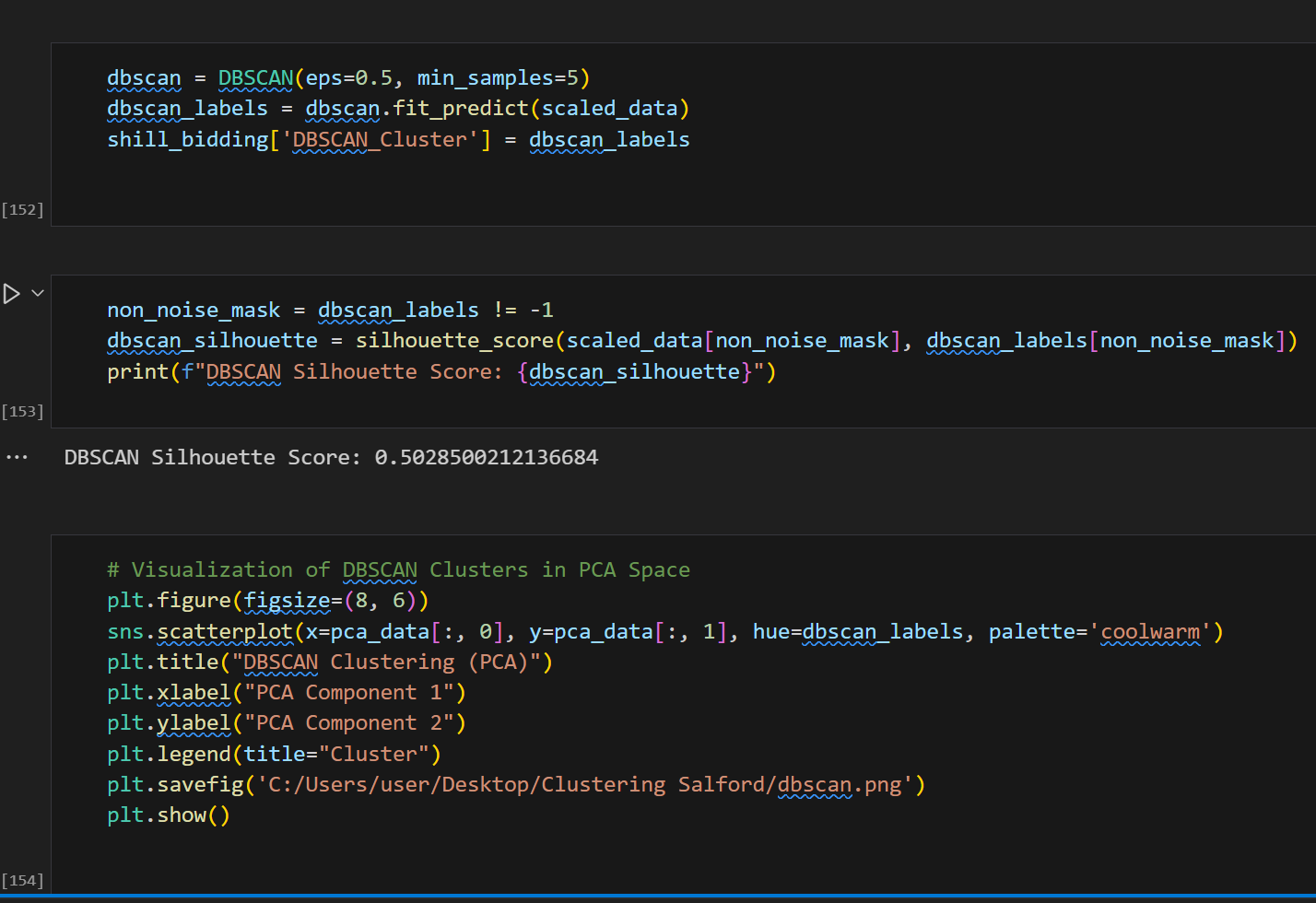
The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) technique was used on the dataset. DBSCAN is a density-based clustering technique that combines closely packed dots and labels outliers as noise. (Ester et al, 1996) The DBSCAN model was used with an eps value of 0.5 and min\_samples set to 5, which are standard values for this approach. The fit\_predict() method was used to create cluster labels for the data, which were then added to the dataset as a new column called DBSCAN\_Cluster. Unlike K-Means, DBSCAN does not require the number of clusters to be provided, making it a more flexible technique for datasets with unknown or changing cluster configurations.



*DBSCAN Clustering*

### 7.1 Silhouette Score for DBSCAN

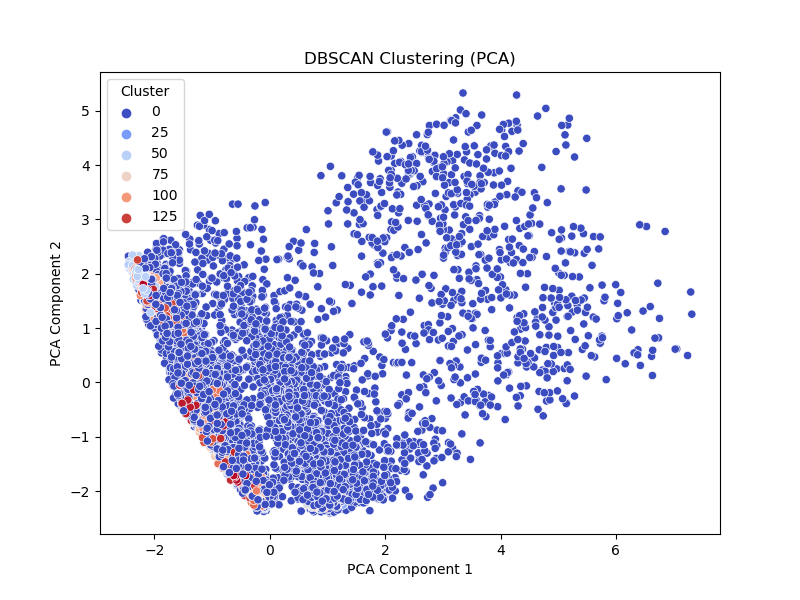
DBSCAN's Silhouette Score was calculated similarly to K-Means (Gusmão et al, 2015). However, because DBSCAN can assign a -1 label to noise points, the silhouette score was calculated exclusively for non-noise points. If there were any noise points, they were removed from the calculations.   
DBSCAN had a silhouette score of 0.50, indicating average clustering quality. This indicated that, while some data points were well-matched to their respective clusters, others may have been more ambiguous in cluster assignment.



*Silhouette Score for DBSCAN*

### 7.2 DBSCAN Cluster Visualization

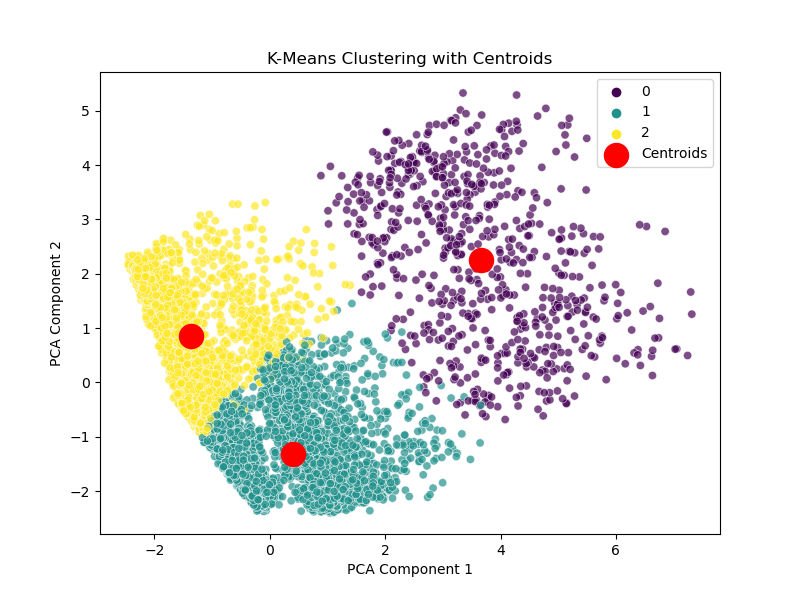
DBSCAN clusters were also visualized in a PCA-reduced 2D space. A scatterplot, similar to the K-Means visualization, was created with different colors denoting the clusters. The visualization demonstrated DBSCAN's ability to detect noise by displaying points labelled -1 independently from the clusters.



*DBSCAN Cluster Visualization*

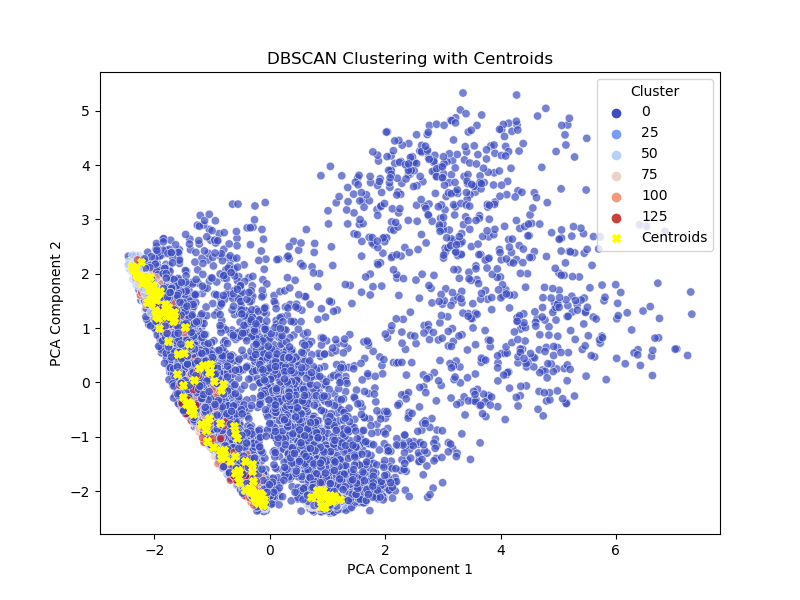
## 8. Visualizing Centroids for K-Means and DBSCAN

K-Means is a centroid-based approach for determining the center of each cluster (Jain, 2010). After K-Means clustering, the centroids were converted to PCA space to see their placements in the 2D projection. The centroids were added to the K-Means cluster visualization as huge red "O" markers. This helped to illustrate the location of each cluster's center in the reduced feature space, allowing for a better comprehension of the cluster distribution (Berkhin, 2006).



*K-MEANS Cluster Visualization with Centroids*

While DBSCAN does not compute centroids, a DBSCAN centroid visualization was created, displaying the results of DBSCAN clustering after reducing the data to two principal components using PCA. The plot's data points were color-coded based on their DBSCAN cluster labels. Different colors were used to symbolize the clusters, making it easy to distinguish between them. The yellow "X" markings represented the centroids of each cluster, calculated as the average of the points in that cluster. These centroids revealed the clusters' central tendency in a smaller 2D space.



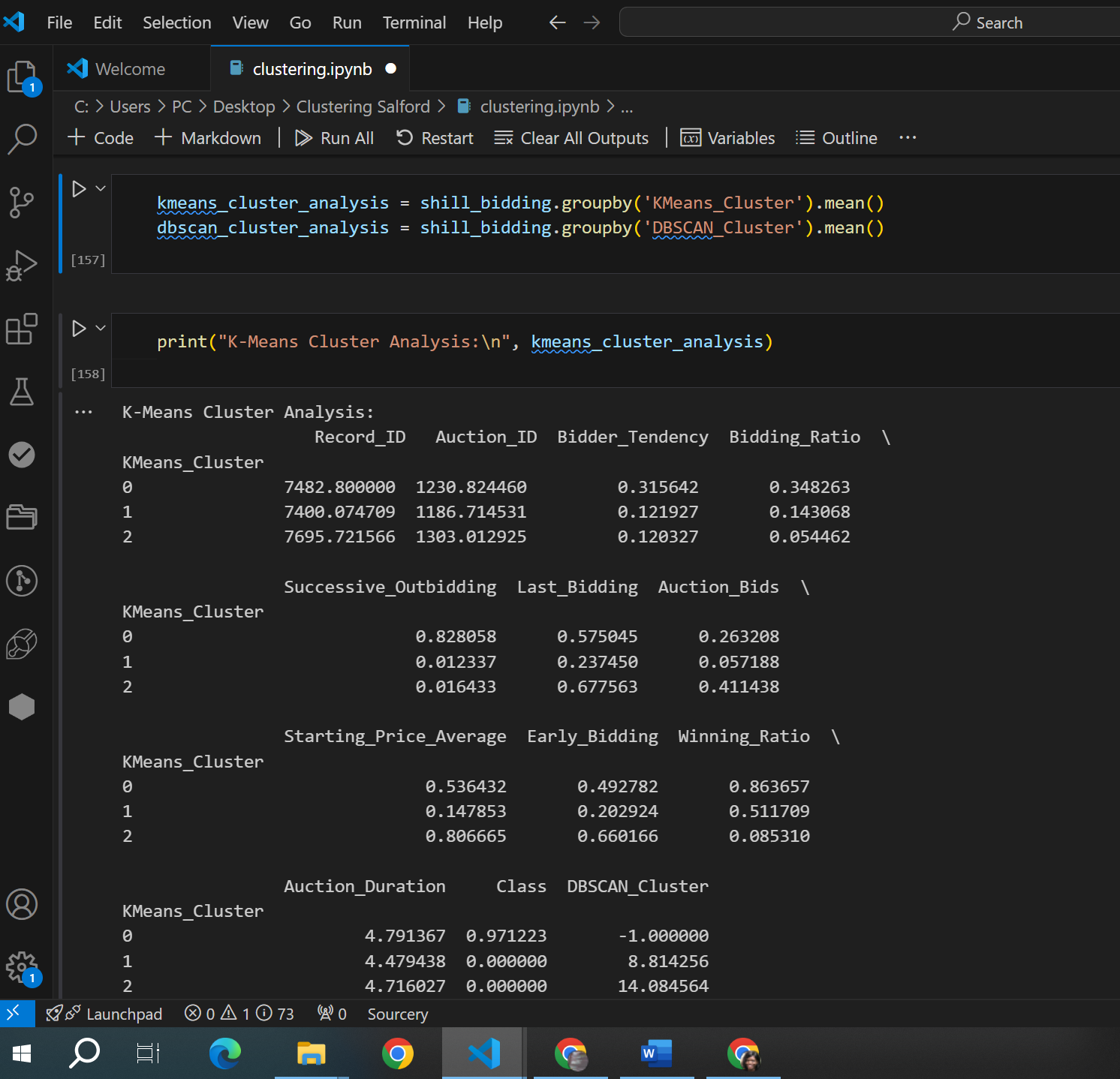
*DBSCAN Cluster Visualization with Centroids*

## 9. Final Cluster Analysis

To acquire a better understanding of the characteristics of each cluster, the average values for each attribute within the clusters were examined (Xu et al, 2005). This was accomplished by categorizing the dataset using the K-Means and DBSCAN cluster labels and calculating the mean of each feature inside each cluster (Ester et al, 1996). The findings were printed and displayed the typical bidding behaviors for each cluster, providing potential insights into the nature of the bids and whether any shill bidding tendencies could be found. (Gusmão et al, 2015)

### 9.1 K-Means Cluster Analysis

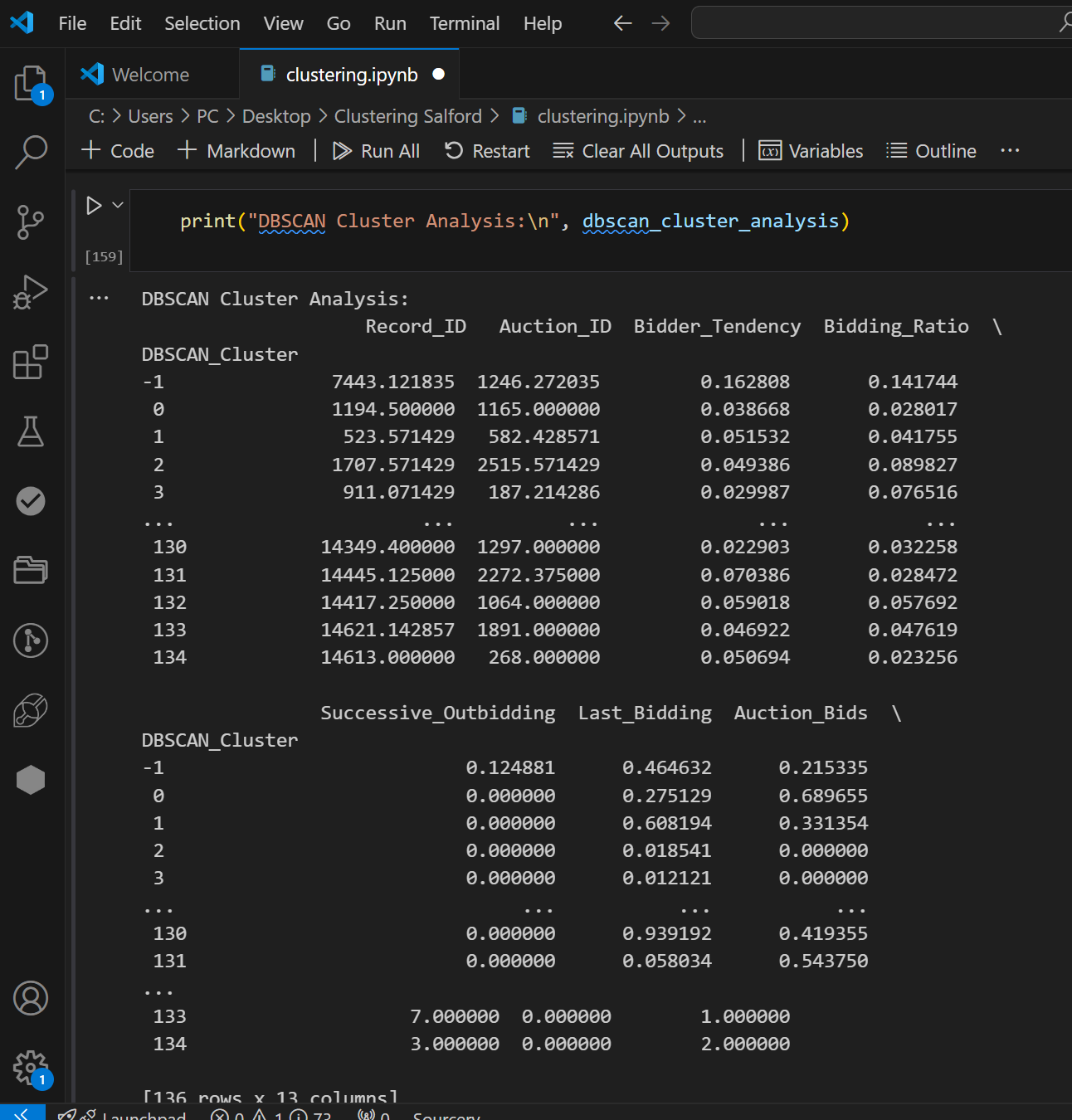
According to the K-Means cluster analysis, there were three different clusters with different traits. Indicating more active and successful bidders, Cluster 0 displayed higher bidder tendency, bidding ratio, successive outbidding, and winning ratio. With lower values for all of these indicators, Cluster 1 showed less active bidders. Although Cluster 2's winning ratio was lower, it showed a modest bidding tendency, indicating more irregular participation. Furthermore, the auction duration was lowest in Cluster 1 and greatest in Cluster 0.



*K-MEANS Cluster Analysis*

### 9.2 DBSCAN Cluster Analysis

Many clusters, including a sizable quantity of noise (Cluster 1), were found by the DBSCAN cluster analysis. Cluster 0, which was characterised by a low bidding ratio and bidder propensity, had greater auction bids, suggesting active participation but less involvement in subsequent bidding. Cluster 1 had a balanced ratio, successful bidding, and moderate bidding behaviour. Minimal bidding activity was seen in Clusters 2 and 3, indicating irregular participation or decreased interest in the auctions. Different clusters' levels of last bidding and successive outbidding indicate that DBSCAN distinguished between regular and irregular bidders.



*DBSCAN Cluster Analysis*

**10. Conclusion**

The Shill Bidding Dataset was analyzed using the K-Means and DBSCAN clustering methods, which yielded important information about the data's structure. Although the features of the clusters varied between the algorithms, both were able to classify the data into unique clusters. (Jain, 2010) (Berkhin, 2006)

While DBSCAN was able to detect noise and manage non-linear cluster forms, K-Means displayed clusters that were more distinct and well-defined. Understanding the clustering results and seeing the patterns in the data was made easier by the clusters' visualization in PCA space (Kriegel et al, 2011).

The Shilling dataset analysis showed different bidding patterns among clusters, with K-Means and DBSCAN offering supplementary information. Three groups with different levels of auction participation were found using K-Means; Cluster 0 had the most activity, while Cluster 2 had the least. DBSCAN, on the other hand, revealed more detailed patterns of bidder behavior, such as active bidders and isolated players (Kriegel et al, 2011), by identifying several clusters and noise. K-Means concentrated on broad patterns, whereas the DBSCAN approach captured more minor fluctuations, including outliers (Han et al, 2012). All things considered, both clustering methods provided insightful viewpoints on bidder patterns, which aided in the comprehension of auction dynamics (Zhang et al, 2014).

## References

1. **Jain, A. K. (2010). Data clustering: 50 years beyond K-means.**  
   *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(7), 645-666.*
2. **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.*
3. **Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms.**  
   *IEEE Transactions on Neural Networks, 16(3), 645-678.*
4. **Gusmão, D. A., & Hruschka, E. R. (2015). A survey of distance-based clustering algorithms: Taxonomy and empirical analysis.**  
   *Computational Statistics & Data Analysis, 87, 1-17.*
5. **Zhang, Z., & Jiang, X. (2014). Clustering large-scale data using DBSCAN and parallel computing.**  
   *Journal of Computational Science, 5(6), 879-886.*
6. **Kriegel, H. P., Kröger, P., & Sander, J. (2011). Density-based clustering.**  
   *In Encyclopedia of Machine Learning (pp. 262-266). Springer.*
7. **Berkhin, P. (2006). A survey of clustering data mining techniques.**  
   *Grouping Multidimensional Data, 25-71.*
8. **Han, J., Kamber, M., & Pei, J. (2012). Data mining: Concepts and techniques.**  
   *Morgan Kaufmann Publishers.*