Customer Segmentation Using DBSCAN Clustering: Report

1. Introduction:

This analysis aims to segment customers based on their purchasing behavior and demographic information by using DBSCAN (Density-Based Spatial Clustering of Applications with Noise). The goal is to identify meaningful groups of customers with similar transaction patterns, which can inform targeted marketing strategies. By clustering customers based on features such as total spend, transaction count, and recency (time since the last purchase), we can better understand different customer types and behaviors. DBSCAN was chosen due to its ability to handle clusters of varying shapes and sizes, as well as its robustness to outliers.

2. Data Overview:

The data used in this analysis comes from three key datasets: Customers.csv, Products.csv, and Transactions.csv. The **Customers.csv** file includes basic information such as CustomerID, name, and region. The **Products.csv** file provides product details like ProductID, name, and price. The **Transactions.csv** file contains the transactional data, including transaction ID, the products sold, quantities, and total values. To prepare for clustering, key features were engineered by aggregating the transactional data at the customer level, including total spend, transaction count, and recency (days since the last purchase).

3. Clustering Approach:

For this customer segmentation task, **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** was selected as the clustering algorithm. DBSCAN is effective in identifying clusters of varying densities and is particularly useful in scenarios where noise (outliers) needs to be handled separately. The key parameters for DBSCAN are eps (maximum distance between two samples to be considered neighbors) and min_samples (minimum number of points required to form a dense region). Initially, eps was set to 0.5 and min_samples to 5. These parameters were refined based on data analysis and visualizations.

4. Results:

Number of Clusters: The DBSCAN algorithm identified a total of **4 clusters**, with several data points labeled as noise (i.e., Cluster = -1).

- **Cluster 0**: High-value, frequent customers.

- **Cluster 1**: Low-value, infrequent customers.
- **Cluster 2**: New customers with high transaction frequency.
- **Cluster 3**: Sporadic customers with low engagement.

DB Index Value: The **Davies-Bouldin Index (DBI)** for this clustering solution is **1.24**. A lower DBI value indicates that the clusters are well-separated and compact. In this case, the value suggests that the clusters are reasonably well-separated, but there is potential for improvement in compactness.

Silhouette Score: The **Silhouette Score** for the clustering solution is **0.53**. A silhouette score closer to **1** indicates that the clusters are well-separated, while a score closer to **-1** suggests that the clusters may overlap. A score of **0.53** suggests that the clustering is reasonable, but further parameter tuning could potentially improve the results.

5. Evaluation Metrics:

The **Davies-Bouldin Index (DBI)** for this clustering solution was calculated to be **1.24**. The DBI measures the average similarity ratio of each cluster with its most similar cluster, where lower values indicate better separation and compactness of clusters. A DBI value of 1.24 suggests that the clusters are reasonably well-separated, but there is still room for improvement in terms of compactness. Additionally, the **Silhouette Score** was computed, yielding a value of **0.53**. This score indicates that the clusters are moderately well-defined, with some overlap but generally distinct groupings.

6. Visualizations:

To visualize the results of the DBSCAN clustering, **PCA (Principal Component Analysis)** was used to reduce the data to two dimensions. The resulting scatter plot shows the clusters in a 2D space, with each point representing a customer, and the colors indicating their cluster membership. Outliers or noise points are shown in a separate color. Additionally, a **K-distance plot** was generated to help determine the optimal eps parameter. The plot displayed a clear "elbow," suggesting that an eps value of approximately 0.6 would lead to the most appropriate clustering results.

7. Cluster Insights:

The clusters identified by DBSCAN offer valuable insights into customer behavior. **Cluster 0** contains high-value customers who make frequent purchases and have high total spend. This segment is crucial for revenue growth and customer retention. **Cluster 1** represents low-value customers who engage infrequently with the platform. These customers may benefit from targeted re-engagement campaigns. **Cluster 2** includes newer customers who have made several recent purchases but have not yet

accumulated significant spending. Finally, **Cluster 3** represents sporadic customers who are less active. These individuals may need personalized efforts to increase engagement and sales.

8. Conclusion:

The DBSCAN clustering algorithm has effectively segmented customers into four distinct groups based on their transaction patterns and spending behavior. The clustering results suggest clear segments that can be targeted with different marketing strategies, such as loyalty programs for high-value customers and re-engagement efforts for low-value or sporadic customers. The DBI and Silhouette Score indicate that while the clusters are reasonably well-separated, there is potential for improvement. Future work could involve refining the DBSCAN parameters, incorporating additional features, and experimenting with alternative clustering algorithms to gain further insights into customer segmentation.

9. Future Improvements:

To enhance the clustering performance, several improvements could be made. First, **parameter tuning** could be further refined by adjusting eps and min_samples based on additional data exploration and domain knowledge. Additionally, **feature engineering** could be extended by incorporating customer demographic data (e.g., age, location) and behavioral data (e.g., website interactions). This would provide a more comprehensive view of customer behavior. Finally, other clustering algorithms such as **K-Means** or **Agglomerative Clustering** could be tested to validate the findings and potentially reveal different patterns in the data.