1. Number of Clusters Formed

The K-Means clustering algorithm has been applied to the data with **4 clusters**. This is specified by setting `n_clusters=4` in the `KMeans` function. The number of customers in each cluster can be seen from the output of `data_agg['Cluster'].value_counts()`.

2. Davies-Bouldin Index

The Davies-Bouldin index (DB Index) is a metric for evaluating the quality of clustering. A lower DB Index indicates better clustering performance, with well-separated and compact clusters. Based on the code:

```
db_index = davies_bouldin_score(scaled_features, data_agg['Cluster'])
print(f'Davies-Bouldin Index: {db_index:.4f}')
```

The calculated DB Index is printed, but since I can't run the code in this environment, you'll need to check the result on your local setup.

3. Other Relevant Clustering Metrics

- **Accuracy Score:** Since you're using the cluster labels as true labels for evaluation (which is generally not recommended for unsupervised learning), the accuracy calculation may not provide meaningful results. Clustering doesn't have true labels, so accuracy isn't a suitable metric here. You might want to focus more on metrics like the **Silhouette Score**, **Adjusted Rand Index**, or **Inertia** (sum of squared distances to the nearest centroid).
- **Silhouette Score** could be another metric for evaluating the quality of clusters, which can be calculated as:
- `from sklearn.metrics import silhouette_score`
- `silhouette_avg = silhouette_score(scaled_features, data_agg['Cluster'])`
- `print(f'Silhouette Score: {silhouette_avg:.4f}')`

4. Visual Representation of Clusters

The code includes a scatter plot to visualize the customer clusters based on `TotalValue` and `Quantity`. The color of each point represents the cluster it belongs to, making it easy to see the distribution of customers within the 4 clusters.

```
plt.scatter(data_agg['TotalValue'], data_agg['Quantity'],
c=data_agg['Cluster'], cmap='viridis')
plt.xlabel('Total Value')
plt.ylabel('Quantity')
plt.title('Customer Clusters')
plt.show()
```

Summary of Evaluation Criteria:

1. **Clustering Logic and Metrics:** The code follows good clustering logic by using K-Means and scaling features before applying the algorithm. However, using accuracy for cluster evaluation isn’t ideal; you might want to consider other metrics like silhouette score.
2. **Visual Representation of Clusters:** The scatter plot effectively visualizes the clusters formed by the K-Means algorithm. The clusters can be easily identified based on the colors, providing a clear view of customer segmentation.