Sales Product Clustering

Shruti Patil
Hof University of Applied Science, Hof, Germany
shruti.patil@hof-university.de

ABSTRACT

Clustering is a data analysis technique where similar items are grouped together based on certain characteristics or features. It aims to uncover hidden patterns and structures within a dataset, organizing data into distinct clusters or groups. The objective of this paper is to understand clustering as an important data mining technique and applying on sales dataset based on their generated revenue. In this paper three algorithms of clustering are applied which include K-means, hierarchical clustering, and DBSCAN. Each method has its own approach of defining similarity and forming clusters. All the approaches are thoroughly explained in the sections below. This paper also aims to assess the clustering results of the applied algorithms to understand the performance of clustering algorithm. Using three different metrics for evaluation, results are assessed and discussed. The study would focus on data exploration, pattern recognition, and decision-making by revealing insights and simplifying complex data structures.

Keywords: Clustering, K-means, DBScan, Hierarchical, Similarity, Cohesion, Dispersion

1 INTRODUCTION

Clustering is an essential data mining tool used for analyzing the big data. Clustering can be defined as a method used to group data points together based on their similarities, with the aim of identifying inherent structures or patterns within the data [22]. Since the beginning, researchers have been working on clustering algorithms to manage their complexity and computational requirements, aiming to enhance scalability and speed. The emergence of big data in recent years has added challenges to this area, prompting the need for more research to improve clustering algorithms. Utilizing data clustering provides insights about data, understanding the patterns in data and making sense of large datasets. It aids in data exploration, visualization, and can also be used for data preprocessing before applying other machine learning techniques. Data clustering follows some key components given by [7], which includes:

(1) pattern representation including feature extraction and/or selection,
(2) definition of a pattern proximity measure appropriate to the data domain,
(3) clustering or grouping,
(4) data abstraction (if needed), and
(5) assessment of output.

These components collectively represent the foundational phases of the clustering process, facilitating the transformation of raw data into cohesive clusters based on similarity or other predetermined criteria. The initial step in data clustering, as mentioned above, involves representing data items as feature vectors or patterns. This involves identifying and defining the features or attributes of the data that are most relevant for clustering. Feature extraction creates new features from raw data, while feature selection chooses the most important existing features. The second step involves calculating the similarity or dissimilarity between data items based on their features. This step involves determining a metric to measure the similarity or distance between data points. Common proximity measures include Euclidean distance, Manhattan distance, and cosine similarity, chosen based on the specific data domain. Third step is the core step where the actual clustering algorithm is applied to the data to group similar data points together. Next step is data abstraction which is applied only if needed to simplify the representation of clustered data. In this step, the clustered data may be summarized or transformed into a more abstract form. This can help in understanding the overall structure and relationships within the data, often by reducing dimensionality or simplifying the clusters. Finally, the results of the clustering are evaluated to determine how well the clustering has performed. This can involve internal validation methods like silhouette score or external validation using ground truth labels, as well as qualitative assessments like visual inspection of clusters. Each component explained here plays a pivotal role in ensuring the overall performance of the clustering endeavor [7]. Performance in clustering analysis can be defined by the ability to correctly group similar data points while maximizing the difference between clusters. Key metrics include cluster cohesion (intra-cluster similarity) and separation (inter-cluster dissimilarity). Effective performance is achieved with high cohesion and high separation.

Clustering can be classified into five categories based on approach for defining cluster. Figure 1 represents different categories of clustering. Partitional clustering methods divide data into non-overlapping clusters, such as K-means, by iteratively optimizing cluster centroids to minimize intra-cluster variance. Hierarchical clustering organizes data into a hierarchical tree-like structure, with each data point initially forming its own cluster and then merging or splitting clusters based on similarity or dissimilarity. Density-based clustering, identifies clusters based on regions of high data density, allowing for irregularly shaped clusters and handling noise effectively. Model-based clustering, like Gaussian Mixture Models, assumes that data points are generated from a mixture of probability distributions, estimating parameters to assign data points to clusters probabilistically. Soft computing clustering methods, such as Fuzzy C-means, assign data points to clusters probabilistically, allowing for partial membership and accommodating uncertainty in cluster assignments [18][16].

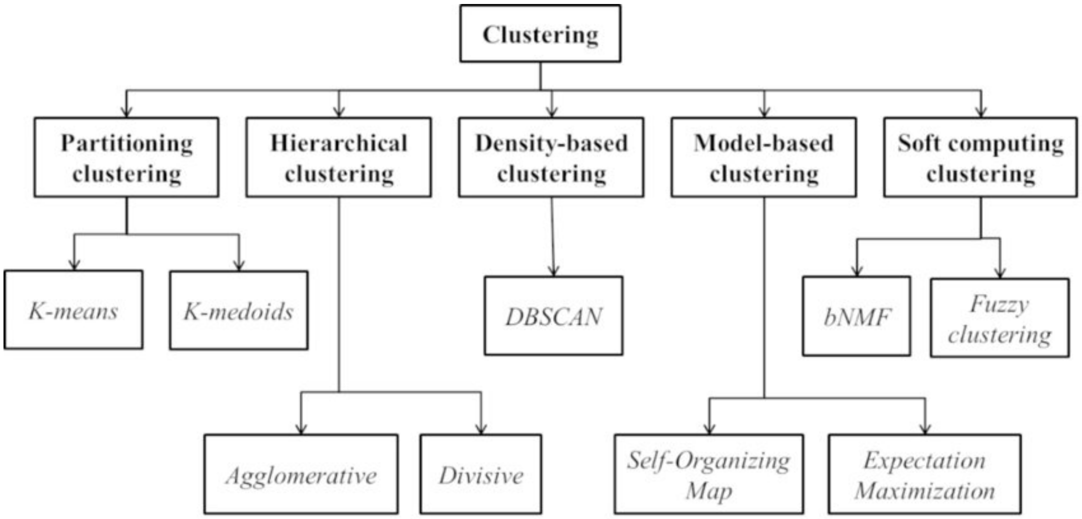


Fig. 1. Types of clustering algorithms [16][18]

2 LITERATURE REVIEW

Cluster analysis is a widely used data mining technique that aims to identify groups or clusters of similar objects based on their attributes or characteristics[9]. It has evolved significantly over the years, with various advancements and improvements being made in the field [10]. One important area of evolution in cluster analysis is the development of different clustering algorithms. These algorithms have become more sophisticated and efficient, allowing for the analysis of larger and more complex datasets. Furthermore, advancements in computing power and the availability of big data have paved the way for the application of cluster analysis in various domains such as marketing, healthcare, finance, and social network analysis. Another area of evolution in cluster analysis is the incorporation of different types of data, including categorical, numeric, and text data [5]. Researchers have also focused on enhancing the interpretability and visual representation of cluster analysis results.

The paper [2] has a problem statement similar to our study. The primary objective of this paper is to propose a data-driven method for clustering retail products based exclusively on customer behavior, deviating from traditional expert-based classification methods. The study aims to demonstrate the effectiveness of this method by formulating the clustering of products as a problem solved using a genetic algorithm, showcasing its behavior in different settings through simulated and real market basket data. By focusing on customer behavior patterns within the market basket data, the study seeks to provide a more objective and automated approach to product categorization, essential for enhancing business decision-making processes in the retail sector [2]. Another research in [1] provides an overview of clustering methods used in Data Mining, starting with defining measures for cluster similarity and dissimilarity. The paper discusses hierarchical, partitional and evolutionary algorithms for clustering. The paper addresses the challenges of clustering in large datasets, highlighting issues such as scalability, computational complexity, and the curse of dimensionality. The paper compares the effectiveness, efficiency, and applicability of three clustering algorithms in different scenarios, providing insights into their strengths and weaknesses [1]. Another review by [8] introduces a simple and efficient implementation of Lloyd's k-means clustering algorithm, termed the filtering algorithm. This algorithm is easy to implement and only requires a kd-tree as the major data structure. The authors establish the practical efficiency of the filtering algorithm through a data-sensitive analysis of its running time. They show that the algorithm performs faster as the separation between clusters increases, indicating its effectiveness in real-world scenarios [8].

Another review in [7] gives an overview of pattern clustering methods from a statistical pattern recognition perspective, aiming to offer valuable advice and references to fundamental concepts accessible to clustering practitioners. The paper discusses important applications of clustering algorithms such as image segmentation, object recognition, and information retrieval, showcasing the practical relevance and impact of clustering in various domains. The authors thoroughly review various literature in field of clustering, highlighting the importance of pattern representation, similarity computation, grouping processes, and cluster representation as key steps in clustering, providing a structured framework for understanding the clustering process.

3 METHODOLOGY

The process commences with the data collection and preprocessing of original data to make it suitable for the study. To ensure that the data is in a suitable format required for the clustering, it is necessary to preprocess the data. Selecting relevant features or extracting meaningful features from the raw data can help improve the clustering process by reducing the dimensionality and focusing on the most informative aspects of the data. Since revenue is the primary feature for the study, informative in this case means fetching columns which might be useful for calculating revenue or which might affect value of revenue. All features required for the study are fetched in new tables for two datasets respectively. Pandas Profiling report was generated to understand our data in depth and find out most correlated dimensions with revenue as primary feature needed for our study. From the report generated and heatmap correlation result it can be said that '*Revenue*' is highly correlated to '*Sold Price*' and '*Ordered Quantities*'. Since, *Product number* column is a mix of alphanumeric and numeric data, it needs to be encoded as clustering expects input data to be numerical so as to avoid data type errors and enhance efficiency. To handle this, new column 'product_id' was generated which consist of auto incremental values, unique for every unique product number. Preprocessing helps clean the data by removing noise, handling missing values, and ensuring consistency, which leads to more reliable clustering outcomes.

On the preprocessed data, appropriate clustering approaches are applied. Clustering approaches was finalized by exploration of diverse approaches to cluster products together based on their revenue along with price and quantity as features. Numerous methods for clustering sample were identified. From many approaches, three approaches were chosen for application due to their versatility across a range of clustering scenarios. These selected approaches are elucidated in Section 5, offering a thorough explanation of their suitability for the study's objectives. K-means can be used with diverse data types, making it versatile clustering algorithm and is also easy to apply and interpret. So first clustering algorithm to be applied is K-means. K-means need a prerequisite of number of cluster 'k' value. To decide appropriate value for number of clusters, Elbow method is utilized. DBScan was the next methodology to be applied due to it's robustness to handle outliers making it suitable for large dataset. Next algorithm applied was hierarchical clustering. This method allows flexible exploration of cluster relationships as it results in a tree-like structure called dendrograms. Due to this hierarchical clustering was applied to see how differently it cluster products than above two methods. All three methods follows different approach to cluster products, hence proving diversity in clustering.

The results from all three methods need to be evaluated using suitable metrics to observe how well the applied algorithms has clustered products.There are two cases for evaluation in cluster analysis, first is when ground truth label are known which means that there is already some expected result which could be later compared with the actual result and second case is when ground truth labels are unknown, in that scenario results can be evaluated using data driven metrics. In our case, ground truth labels are unknown, hence data based evaluation metrics could be utilized. Three metrics are selected to evaluate results which are Silhouette score, Calinski Harabasz score and Davies Bouldin index. After evaluating the results using these three metrics, the results are visualized and compared to understand which algorithm performed well for given sample dataset. All mentioned approaches and metrics are briefly explained in further sections. Figure 2 illustrates the methodology explained in this section.

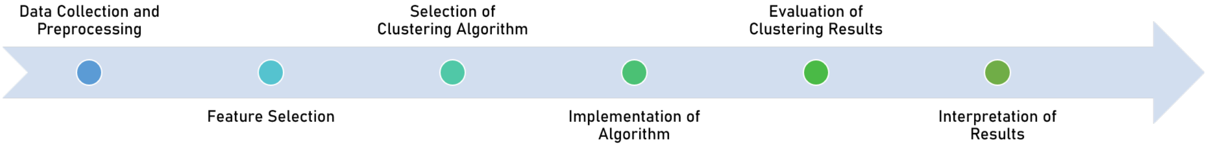
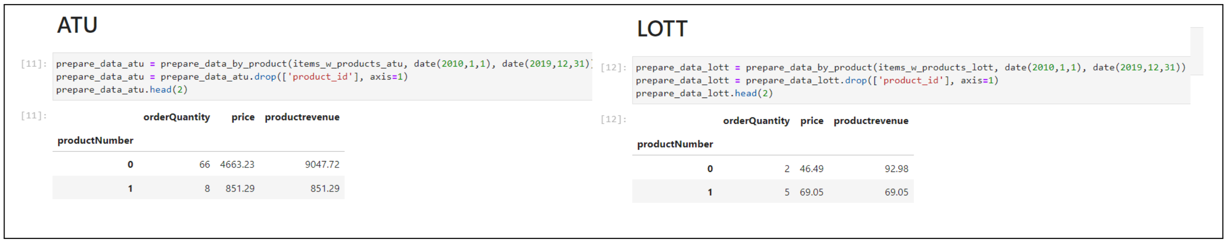

Data Collection and Preprocessing
Feature Selection
Selection of Clustering Algorithm
Implementation of Algorithm
Evaluation of Clustering Results
Interpretation of Results

Fig. 2. Methodology used for the study

4 DATA PREPARATION

The database used for the study is PostgreSql. The dataset under consideration captures essential details regarding Product Sales for two datasets from two companies ATU and LOTT which are both provided from "Speed4Trade". For study, two new materialized views were created for ATU and LOTT respectively, combining all necessary dimensions from original dataset. Since study objective is to cluster products based on revenue as a primary feature, a new column was created which stored the calculated revenue value by multiplying order quantity and number of products sold from existing dataset making it easier to preprocess revenue data further. This would eliminate the hassle of repeatedly calculating revenue. Null values, empty data and missing data is handled in next step by removing all records from original data as it could negatively impact the clustering results. Since most of the clustering approaches rely on distance calculations between data points, missing values can distort these calculations, leading to inaccurate cluster assignments. Hence, considering this data would not make any significant difference in clustering result, so it is better to delete these records and consider remaining records with complete details available. Since product number column have alphanumeric values, two another tables were created for ATU and LOTT respectively to store all product list referenced by unique numerical value. All above steps are directly applied on database tables. Now data can be fetched and stored in Pandas Dataframe making efficient usage and could be modified as per the requirement. Functionality was created to prepare product data such that it results in total revenue value of every product for given time frame. Utilizing this functionality, data is filtered as per the given time frame and grouped by product number with total revenue, total price and total ordered quantity values as parameters. Index is set to product number. Final data output is shown in Figure 3 for ATU and LOTT respectively. For each product number, total order quantity, total price and total revenue is calculated for time frame 2010 to 2019.



ATU

```
[11]: prepare_data_atu = prepare_data_by_product(items_w_products_atu, date(2010,1,1), date(2019,12,31))
      prepare_data_atu = prepare_data_atu.drop(['product_id'], axis=1)
      prepare_data_atu.head(2)
```

```
[11]:                 orderQuantity    price   productrevenue
      productNumber
                  0            66  4663.23          9047.72
                  1             8   851.29           851.29
```

LOTT

```
[12]: prepare_data_lott = prepare_data_by_product(items_w_products_lott, date(2010,1,1), date(2019,12,31))
      prepare_data_lott = prepare_data_lott.drop(['product_id'], axis=1)
      prepare_data_lott.head(2)
```

```
[12]:                 orderQuantity  price  productrevenue
      productNumber
                  0             2  46.49           92.98
                  1             5  69.05           69.05
```

Fig. 3. Preprocessed Data (2010-2019) for Cluster Analysis

5 IMPLEMENTATION

This section elaborates on all algorithms used for the study, outlining their methodologies, application in the research, and respective pros and cons for analysis.

5.1 K-Means

K-means clustering is a fundamental distance based unsupervised machine learning algorithm used to partition a dataset into K distinct clusters based on similarity. It is a partition based clustering algorithm employing squared error criterion[7]. The K-Means algorithm clusters data by aiming to divide samples into n groups with equal variance, while minimizing a metric called inertia or within-cluster sum-of-squares[17]. The methodology involves iteratively assigning data points to the nearest centroid and updating centroids to minimize the within-cluster sum of squared distances. This algorithm requires number of clusters k to be specified as a prerequisite. Various methods are available to find optimal value for number of cluster k. Initially, each sample is assigned to its nearest centroid. Then, new centroids are generated by calculating the mean value of all samples assigned to each previous centroid. The algorithm iterates these steps until the difference between old and new centroids is below a specified threshold, indicating minimal movement of centroids. The means are commonly called the cluster "centroids"[17].

K-means has been utilized for this study due to it's simplicity, computational efficiency and scalability making it a good choice for large dataset. However, its effectiveness depends on the appropriate choice of K, the number of clusters, which may not always be known beforehand. Additionally, K-means can be influenced by the initial choice of centroids and may converge to local rather than global optima. However, despite these challenges, K-means is widely used across diverse domains such as image segmentation, customer segmentation, anomaly detection, and document clustering. Next section describes how k-means is applied for product clustering based on revenue.

5.1.1 Use Case - Product Clustering.

The very first step is to import K-means from sklearn library. The initial requirement to start with an algorithm is to decide the optimal number of cluster k value. For which, Elbow method also referred as Elbow curve is utilized. Elbow method is the graphical interpretation of sum of squared error (SSE) for various number of k's. SSE is the sum of the average Euclidean Distance of each point against the centroid[14]. It is the most popular method to find optimal value of k. This method is quite straightforward and easily interpretable. Due to it's graphical interpretation, value for k can be visually identified by recognising the point where adding more clusters does not significantly decrease the within cluster sum of squares. In other words, when the value drops drastically and forms a smaller angle, then the value of k is found. Starting from minimum value for k as 1 and looping till maximum value 10, SSE value is calculated. Higher value for k indicates more clusters and having too many clusters can make the results difficult to understand and use. When adjusting the value of k, the newly formed cluster tends to resemble the previous one, or the alteration in errors remains relatively unchanged, leading to the chosen value of k. For given sample, elbow plots are generated for both datasets and results looks like in Figure 4.

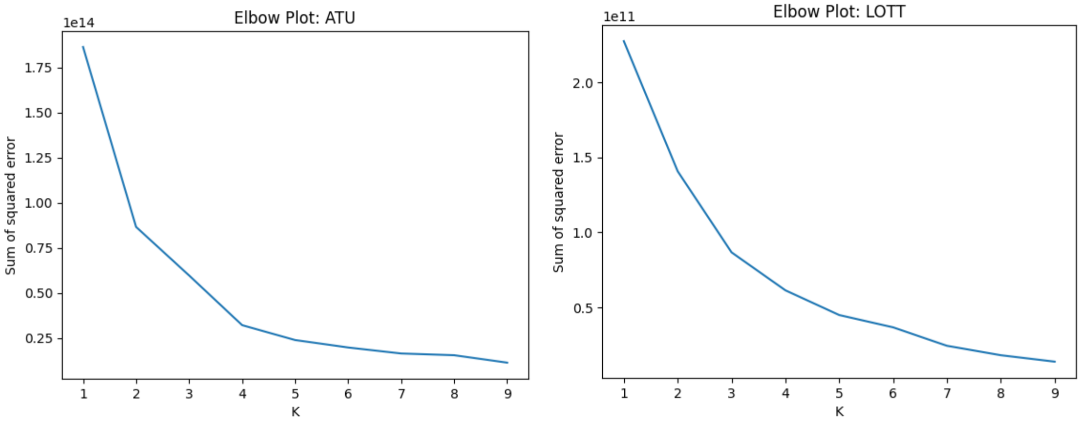
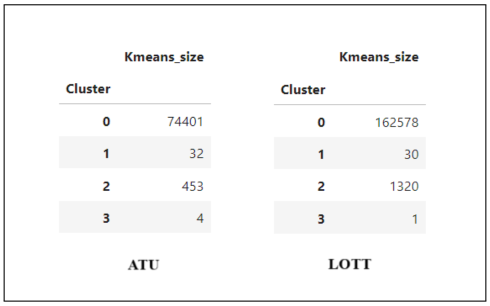


Fig. 4. Elbow Plot for ATU and LOTT

From the elbow plots in Figure 4, optimal value for k is 4 or 5 for ATU and 4 to 6 for LOTT. It can be observed that after k=4, there is minor difference in values for SSE and also visually smaller angle can be observed. Hence, k=4 would be optimal value for application for both datasets. Using this value for k, K-means is applied on sample and results for clustering are visualised using a scatter plot. Figure 5 depicts the size of clusters obtained using K-means clustering. The scatter plot results are depicted in Figure 6 and 7 for ATU and LOTT datasets respectively. Both figures have 2 scatter plots depicting the correlation between revenue with order quantity and revenue with price.

From Figure 5 reveals that in both ATU and LOTT dataset, the majority of data points are in Cluster 0, while Cluster 2 contains a moderate number of data points. Clusters 1 and 3 have significantly fewer data points, with

Cluster 3 having the least. In Figure 6 and 7, first scatter plots representing revenue with order quantity, it is noticeable that in both datasets, clusters are primarily determined by revenue, with order quantity values having minimal impact on cluster formation. However, in second scatter plots representing revenue with price, a degree of linearity emerges in clusters where both dimensions contribute equally to cluster formation. In both scenarios, clusters are organized based on increasing revenue values, from low to high. The evaluation of these results will be conducted in Section 6 using three different metrics.



	Kmeans_size
Cluster	
0	74401
1	32
2	453
3	4

ATU

	Kmeans_size
Cluster	
0	162578
1	30
2	1320
3	1

LOTT

Fig. 5. Size of clusters in ATU and LOTT using K-Means

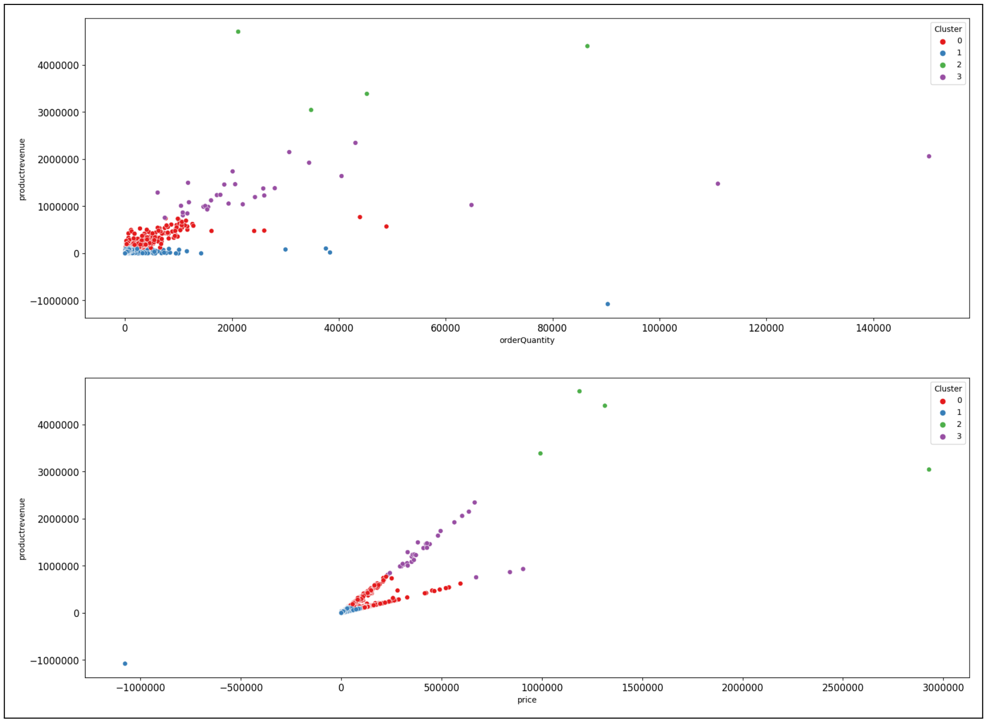


Fig. 6. Four clusters obtained using K-Means on ’ATU’ dataset with three features

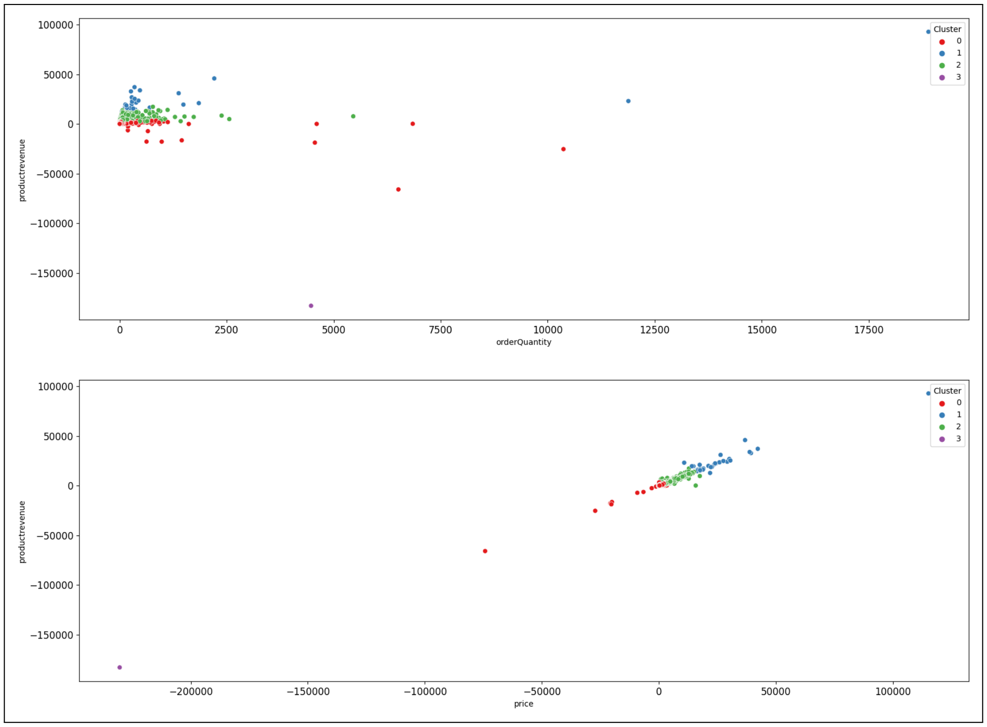


Fig. 7. Four clusters obtained using K-Means on 'LOTT' dataset with three features

5.1.2 Advantage and Disadvantages.

Based on it's application and usage in above section some advantages and disadvantages could be listed as follows:
Advantages:

(1) Simple and easy to implement due to its straightforward iterative clustering updates to assign data points to cluster.
(2) Efficient for large datasets due to its computational scalability.
(3) Works well with datasets where clusters are spherical or roughly convex in shape.This is because the score assumes that clusters are separated by clear boundaries and that points within a cluster are close to each other.

Disadvantages:

(1) Requires the number of clusters (k) to be specified beforehand, which can be challenging to determine.
(2) Sensitive to the initial placement of centroids, which can cause the algorithm to settle on a less-than-best solution.
(3) Assumes clusters have equal variance, which might not be the case in real-world datasets.

5.2 DBScan

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density based clustering method which clusters samples based on the density, such that datapoints in a cluster have high density with each other as compared to datapoints in different cluster. DBScan is a non-parametric approach where the groups in the data are considered high density areas[19]. Density based clustering identifies clusters of arbitary shapes and sizes with seperated outliers[20]. DBSCAN automatically determines the number of clusters without requiring a predefined value, distinguishing it from K-means[19]. Even though this method do not require number of cluster value to be specified, it needs *Eps* and *MinPts* values as prerequisite to calculate number of clusters. This method is very robust to outliers and noise in sample dataset, making it suitable for large datasets.

Since DBScan clusters data based on density, it is important to clarify what density means in this context. The parameter '*Eps*' refers to the maximum distance between two points to be considered neighbors, and '*MinPts*' is the minimum number of points required to form a dense region within this distance. A "dense region" is thus defined by having at least MinPts within an Eps radius. Points within the Eps radius that do not meet the MinPts requirement are labeled as "border points," while any remaining points are classified as noise or outliers.

Illustration in Figure 8, with MinPts=3 red points (labeled D) represent those within a dense region, each having at least three neighbors within the distance Eps. Green points (labeled B) are border points, having a neighbor within Eps but fewer than three. The blue point (labeled O) is identified as an outlier since it lacks neighbors within the specified distance Eps [11].

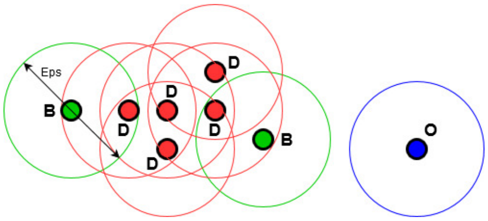


Fig. 8. Illustration of different points with Eps and MinPts=3 in DBScan [11]

5.2.1 *Use Case - Product Clustering.*

Utilizing DBScan library from sklearn, clustering is applied on sample dataset. To determine appropriate values for hyperparameters *Eps* and *MinPts*, heatmap is generated along with manually trying and testing multiple values of Eps and MinPts for both datasets. Heatmap generated is illustrated in Figure 9. From the heatmap result and manually testing the values, Eps = 190 and MinPts = 24 was finalized for ATU and EPS = 60 and MinPts = 9 for LOTT for the study. These values were decided based on the Silhouette Score it generated. Highest score indicated better values. Silhouette Score method will be briefly described in Section 6. DBScan algorithm is applied on sample dataset. Figure 10 depicts size of clusters obtained by DBScan algorithm.

DBSCAN generated five clusters along with an outlier cluster (-1) for ATU, and six clusters with outliers as separate cluster for LOTT. The cluster sizes show considerable variation, with cluster 0 containing the highest number of observations in both cases, while the other clusters have relatively fewer observations. There are 7481 outliers for ATU and 1661 outliers for LOTT. Two scatter plots are presented for revenue with order quantity and price respectively, displaying the results illustrated below in Figure 11 and 12 for ATU and LOTT respectively.

The data depicted in Figure 11 and 12 indicate that the clusters formed exert a notable impact on revenue with order quantity, and there appears to be a linear relationship between revenue and price as clustering progresses from the lowest to the highest revenue values. The outliers are clustered separately, and because of their high number, they overshadow the points belonging to other clusters. The reason for high number of outliers can have many reasons. Choice of parameters 'eps' and 'minPts' are crucial, changes in these values might change size of clusters and outliers. Also dataset used might make the results differ.

Figures 13 and 14 present the same clustering results produced by the DBScan algorithm. These figures offer a slightly zoomed-in view and exclude any outliers. The clusters labeled with different colors in both plots appear to be somewhat scattered. Clusters seem to be elongated and irregular, indicating that they might not follow a simple convex shape, but they cannot be categorized as concave as clusters can be clearly separated from each other. Cluster 0 is the largest and most spread out. The points are widely dispersed, forming irregular shapes. Other clusters are smaller in size and compact in nature. This might be a cause of DBScan's high computational complexity and hence limiting its ability to handle large datasets.

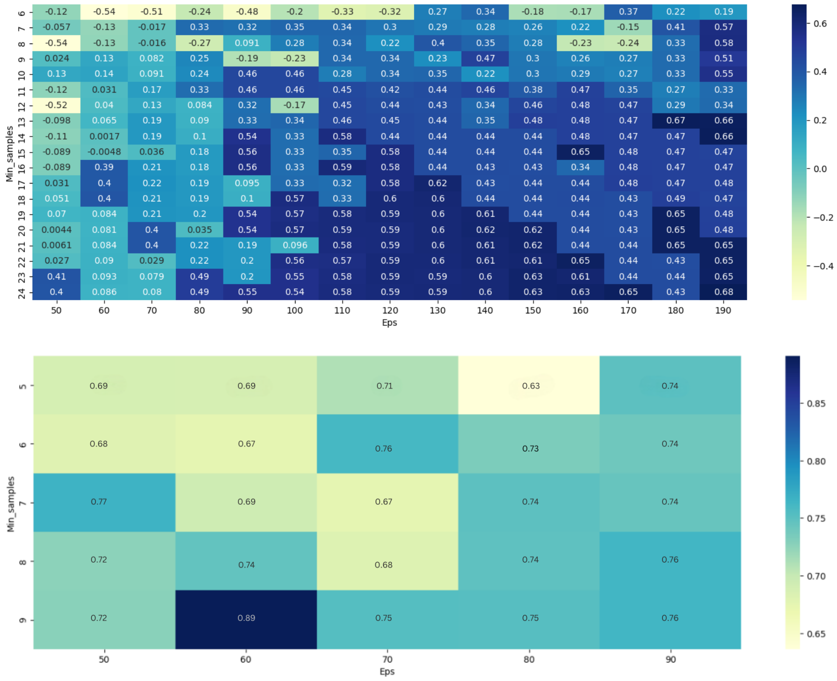
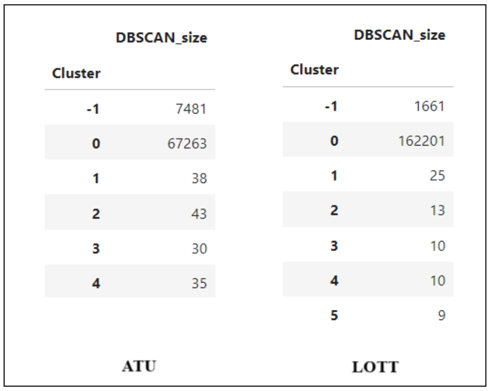


Fig. 9. Heatmap generated for ATU and LOTT



	DBSCAN_size
Cluster	
-1	7481
0	67263
1	38
2	43
3	30
4	35

ATU

	DBSCAN_size
Cluster	
-1	1661
0	162201
1	25
2	13
3	10
4	10
5	9

LOTT

Fig. 10. Size of clusters in ATU and LOTT using DBScan

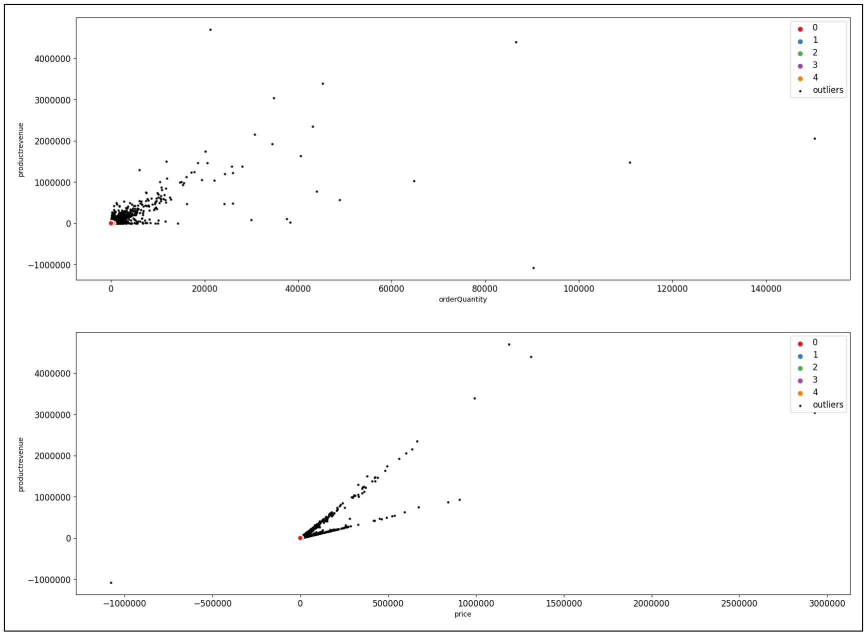


Fig. 11. Six clusters obtained using DBScan on 'ATU' dataset with three features

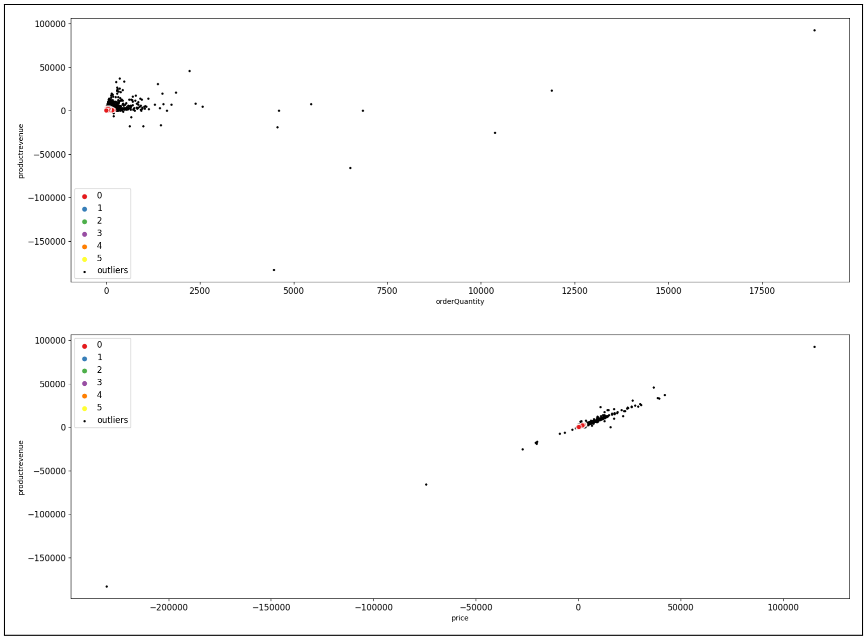


Fig. 12. Seven clusters obtained using DBScan on 'LOTT' dataset with three features

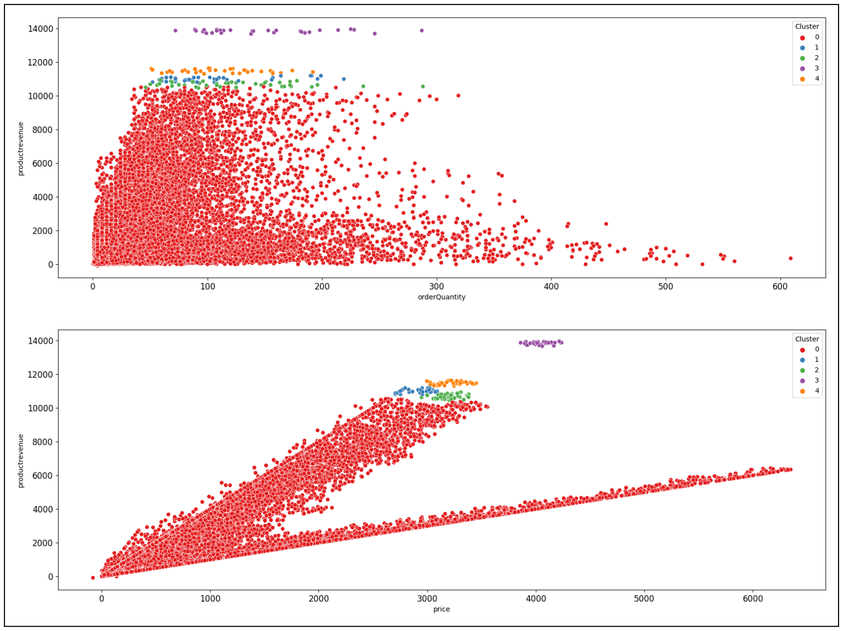


Fig. 13. Clusters formed by DBScan excluding outliers on 'ATU' dataset

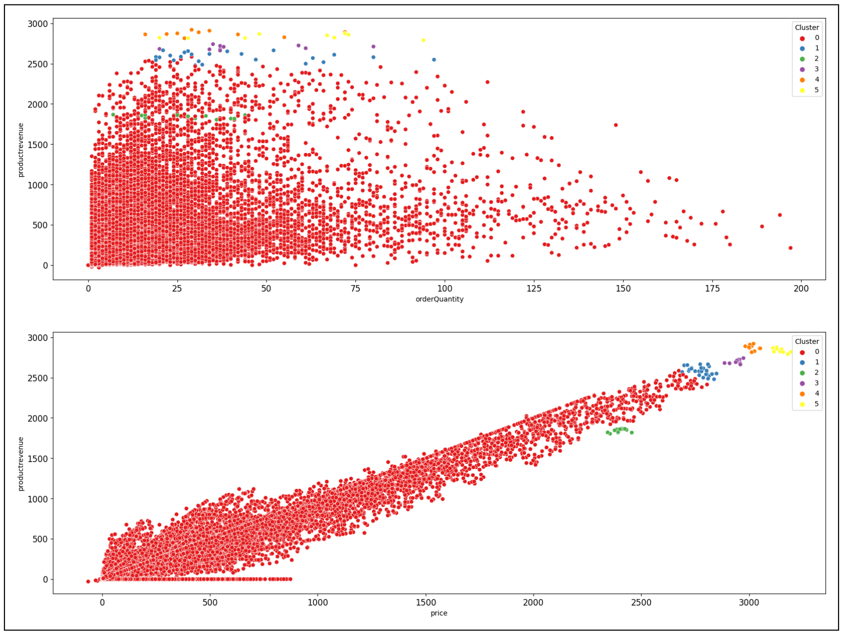


Fig. 14. Clusters formed by DBScan excluding outliers on 'LOTT' dataset

5.2.2 Advantage and Disadvantages.

As elaborated previously, DBSCAN clusters data points by density. Also it's ability to handle noise and discover clusters of varying densities makes it a versatile tool for many applications. Below are some of its advantages and disadvantages:
Advantages:

1. Automatically determines the number of clusters based on the density of data points.
2. Robust to outliers and noise in data .
3. Not sensitive to the initial selection of parameters, such as distance threshold (Eps) and minimum points (MinPts).
4. Handles irregularly shaped clusters.

Disadvantages:

1. Requires careful selection of parameters, such as Eps and MinPts, which can significantly impact the clustering results.
2. Computationally expensive for large datasets, especially when using the Euclidean distance metric.
3. Struggle with high-dimensional data due to the curse of dimensionality, leading to reduced clustering performance.

5.3 Hierarchical

Hierarchical clustering is a clustering method which divides a dataset into sequence of nested partitions called as dendrograms[13]. The functionality of a hierarchical clustering algorithm is demonstrated through the utilization of the two-dimensional dataset showcased in Figure 15. Within this illustration, seven datapoints denoted as A, B, C, D, E, F, and G are distributed among three clusters. Through the hierarchical algorithm, a dendrogram is generated to depict the hierarchical arrangement of patterns and the corresponding similarity thresholds where alterations in groupings occur. A dendrogram corresponding to the seven points in Figure 15 utilizing single-link algorithm is shown in Figure 16 [7][6].

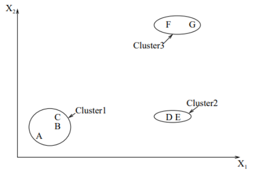


Fig. 15. Points falling in three clusters [7][6]

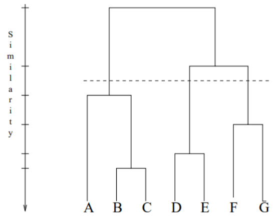


Fig. 16. The dendrogram obtained using single linkage algorithm [7][6]

Hierarchical clustering have diverse variants like ward linkage, single linkage, complete linkage and average linkage. These algorithms differ in how they define cluster similarity. In ward-linkage method, it tries to minimize the sum of squared difference within all clusters similar to k-means algorithm. In the single-link method, cluster distance is the minimum distance between any pair of patterns from different clusters, while in the complete-link algorithm, it's the maximum distance. Average linkage minimizes the mean between all observations of pair of clusters. Despite this difference, both algorithms merge clusters based on minimum distance criteria [7][6][17]. Hierarchical clustering are subdivided into agglomerative and divisive hierarchical algorithm. Agglomerative algorithm assumes every single sample point as one cluster and then it replicate merging of closest cluster together until all data are in one cluster. This algorithm follows a bottom-up approach of merging. In contrast, divisive algorithm follows top to bottom approach where all sample points are considered as a part of one single big cluster and then it repeats splitting of points into smaller cluster [13].

5.3.1 Use Case - Product Clustering.

In study's scenario, Agglomerative hierarchical library is utilized from sklearn. This algorithm can handle non-linear cluster shapes and can scale to large number of samples making it a good choice for the analysis. Linkage method is used to create dendrogram for a sample dataset. It has three parameters, sample dataset, linkage method type and metric type. For study, linkage method selected is ward linkage as it aims to minimize sum of squared differences similar to k-means. This will aid to understand differences between these two methods and how differently it clustered data considering same distance metrics. Figure 17 depicts dendrograms produced.

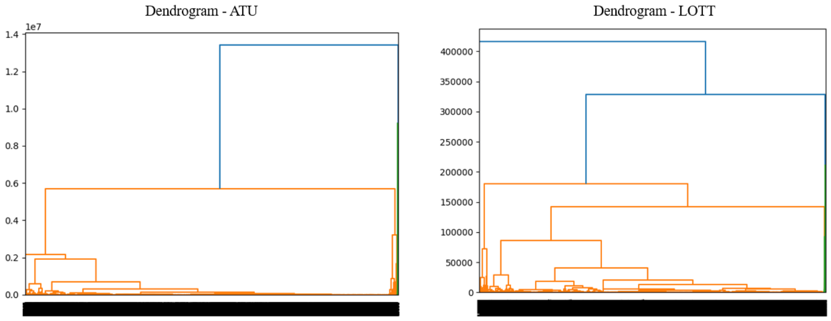
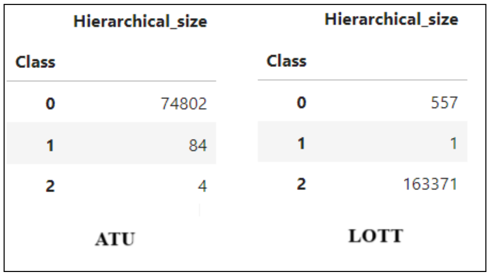


Fig. 17. Dendrogram produced for ATU and LOTT respectively



Class	Hierarchical_size
0	74802
1	84
2	4

ATU

Class	Hierarchical_size
0	557
1	1
2	163371

LOTT

Fig. 18. Size of clusters in ATU and LOTT using Hierarchical

From the resulting dendrogram in Figure 17, it is observed that dendrogram has divided dataset into 3 clusters. Hence, number of cluster value for cluster analysis is decided to be 3. Agglomerative clustering is applied with ward linkage method. The clustering results are then depicted using a scatter plots. Figure 18 depicts size of clusters obtained by Hierarchical clustering algorithm. For ATU dataset, Cluster 0 has highest data points, followed by cluster 1 and cluster 2 having lowest data points. While in LOTT dataset, Cluster 2 has the highest data points, followed by cluster 0 and cluster 1 having lowest data points. Figure 19 and 20 represents hierarchical algorithm results for both datasets.

The outcomes derived from the hierarchical algorithm exhibit a notable similarity in the grouping of products when compared to the results obtained from K-means and DBScan methodologies. Unlike K-means and DBScan, this particular algorithm does not isolate outliers into separate clusters. Instead, it integrates them into the overall clustering structure. Nevertheless, the primary determinant in forming these clusters remains the revenue value associated with each product. In contrast, the influence of order quantity on the clustering process is relatively insignificant. Furthermore, a discernible pattern emerges where there is a linear relationship between the price and revenue values of the products, suggesting a consistent trend across the dataset.

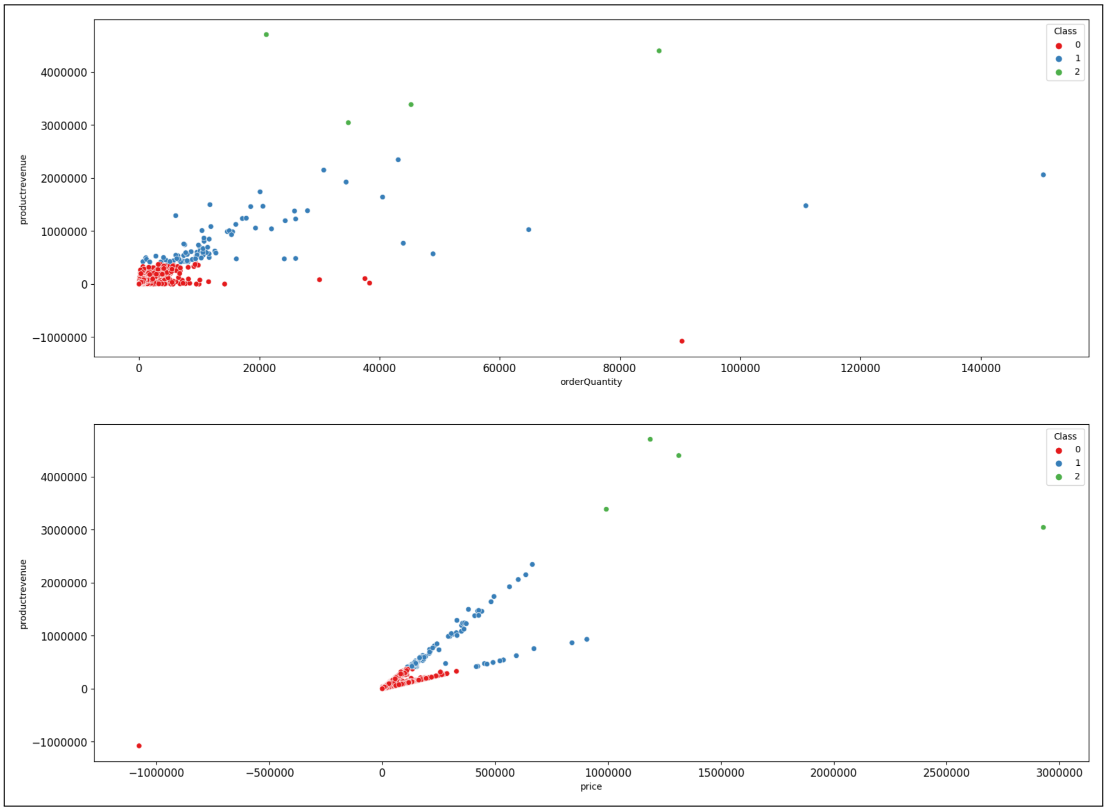


Fig. 19. Three clusters obtained using Hierarchical on 'ATU' dataset with three features

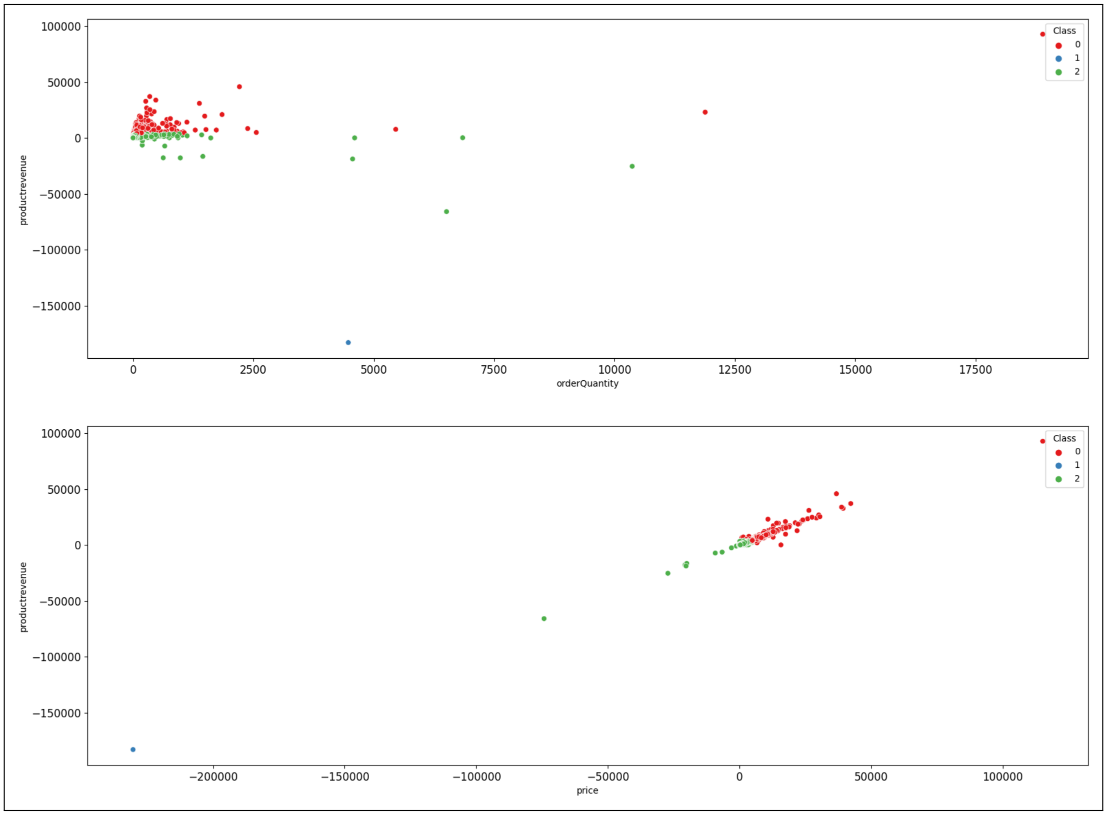


Fig. 20. Three clusters obtained using Hierarchical on 'LOTT' dataset with three features

5.3.2 Advantage and Disadvantages.

As previously mentioned, hierarchical clustering offers a range of variants and customization options, allowing for flexibility in clustering approaches through modifications to linkage methods and distance metrics. Here are some advantages and disadvantages of the algorithm, highlighting its distinctiveness from methods such as K-means and DBSCAN.
Advantages:

(1) Provides a hierarchical structure of clusters using dendrograms.
(2) Can handle non-linear cluster shapes and clusters of varying sizes.
(3) Can use various distance metrics to measure similarity between data points.

Disadvantages:

(1) Computationally intensive, especially for large datasets, due to its quadratic time complexity.
(2) The choice of linkage method (e.g., single, complete, average) can significantly impact the clustering results.
(3) Memory-intensive for large datasets can limit the scalability.
(4) Sensitive to noise and outliers in the data, which can affect the structure of the dendrogram.

6 EVALUATION

This section assesses the clustering performance of all the methodologies applied in clustering the products based on their revenue values. Here performance refers to the effectiveness with which the algorithm identifies and groups similar data points into clusters. Performance is evaluated based on several criteria, including the coherence within clusters (how similar the data points are to each other within the same cluster) and the separation between clusters (how distinct or different each cluster is from the others). High-performance clustering results in groups that are internally cohesive and well-separated from each other, effectively capturing the underlying structure of the data. To accomplish this, three evaluation metrics are employed varying in evaluation method: Silhouette score, Calinski-Harabasz Score and Davies-Bouldin Index. This portion provides detailed explanations of these metrics and presents the scores for all the methods applied to the sample dataset.

6.1 Silhouette Score

As outlined in Section 3, since ground truth labels are unavailable for the sample dataset, a data-driven metric is utilized to assess clustering results. The advantage of the Silhouette score lies in its independence from a training set for evaluation, rendering it a suitable choice for clustering tasks[23]. The Silhouette score function *sklearn.metrics.silhouette_score* given by Scikit-learn calculates the average silhouette coefficient across all samples [21][17]. Silhouette coefficient is calculated using the mean of intra cluster distance 'a' and mean of nearest cluster distance 'b' for each sample [17]. More clearly 'a' is the mean of distances between sample and all other points in same cluster and 'b' is the mean of distance between sample and all other points in next nearest cluster[15]. Silhouette coefficient of a sample is given by $(b - a)/max(a, b)$ [17]. A silhouette score close to +1 indicates that the data point is appropriately placed within its cluster. A score near 0 suggests that the data point could potentially belong to another cluster. A silhouette score around -1 implies that the data point is likely in the wrong cluster [21]. Higher the score, better are the results [15]. Hence an outstanding clustering configuration will have an average silhouette score near +1, meaning that most data points are well matched to their own clusters and poorly matched to neighboring clusters. This suggests strong intra-cluster similarity and clear inter-cluster separation [15].

The silhouette coefficient is a versatile metric that doesn't rely on the convexity of clusters. It measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). This makes it applicable to a wide variety of clustering shapes, including concave clusters. It provides a quantitative measure of how well each data point fits into its cluster. Silhouette scores allow you to compare the performance of different clustering algorithms, which might produce clusters of varying shapes. Hence, this metric seems to be a good choice for evaluating the results from algorithms applied for the study.

The calculated Silhouette score for sample data is shown in Figure 21 and 22 for ATU and LOTT datasets respectively. In the ATU dataset, Hierarchical achieves the highest silhouette score, nearly reaching 1 at 0.9891, followed by k-means clustering with a score of 0.9686. DBScan, although lower as compared to other methodologies, still obtains a respectable score of 0.6769 overall as gained result is still above an average of 0.5. Additionally, in the LOTT dataset as well, hierarchical clustering boasts the highest silhouette score of 0.9693. K-means follows closely with a score of 0.9504, while DBScan achieves a commendable score of 0.8909, indicating effective clustering overall. The high silhouette scores indicated strong clustering performance on the sample data. This metric confirms that the clusters are well-defined and distinct. The clustering algorithm effectively groups similar data points while separating different ones. High silhouette values close to 1 signify that the clustering is highly effective. Overall, the evaluation shows that the clustering approach is robust and reliable.

6.2 Calinski-Harabasz Score

Calinski-Harabasz score is another evaluation metric for cluster analysis. This function is a score defined as ratio of sum of between cluster dispersion and sum of within cluster dispersion [17][12]. Dispersion refers to the degree of spread or variability in a dataset[3]. Calinski-Harabasz Score *CH(K)* is given by [25],

$$CH(K) = \frac{B(K)(N-K)}{W(K)(K-1)}$$

where, K is corresponding number of cluster, N is the size of dataset, B(K) is between cluster dispersion and W(K) is within cluster dispersion. B(K) measures how well the clusters are separated from each other (the higher the better) and W(K) measures compactness of clusters (the smaller the better) [25].

Calinski-Harabasz score, also referred as Variance Ratio Criterion[17], evaluates the clustering quality based on the ratio of the sum of between-cluster dispersion to within-cluster dispersion. This makes it a useful measure of how well-defined clusters are, regardless of their shape. The CH score does not assume a specific shape for the clusters. It assesses the overall compactness and separation of clusters based on variance, making it suitable for evaluating clusters of any shape, including non-convex clusters. The CH score is computationally efficient, making it suitable for evaluating large datasets where clusters might have complex shapes. Its efficiency allows for quick assessments and comparisons across different clustering results. A high CH score means that the ratio of the sum of between-cluster dispersion and within-cluster dispersion is large. This indicates that the clusters are well-separated and internally tight.

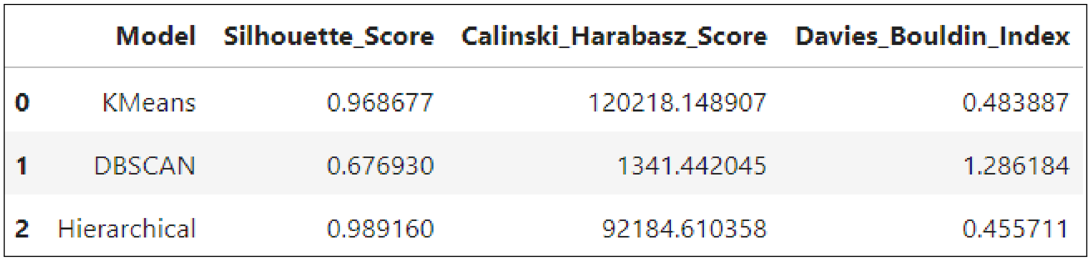
Using scikit learn's function *sklearn.metrics.calinski_harabasz_score* for application, a score is returned over all samples. The greater the ratio, the higher the value of the CH index, indicating a more effective clustering outcome. The result score for sample data is depicted in Figure 21 and 22. In the ATU dataset, K-means achieves the highest CH score of 120218, followed by hierarchical clustering with a score of 92184. DBScan lags behind with a score of 1341. In contrast, for the Lott dataset, K-means obtains a best score of 151355, followed by hierarchical clustering with score of 133234, while DBScan achieves a notably lower score of 5129.

6.3 Davies-Bouldin Index

Scikit Learn's function *sklearn.metrics.davies_bouldin_score* calculates the davies bouldin index for sample dataset. DB Index is an average similarity measure of each cluster with it's most similar cluster [17]. The aim of using the Davis-Bouldin Index for measurement is to simultaneously maximize the distance between clusters while minimizing the distance between points within each cluster [24][4]. For each cluster i, the DB index computes the average distance between each point in the cluster and the centroid of the cluster. It also calculates the distance between the centroids of cluster i and the centroids of all other clusters.

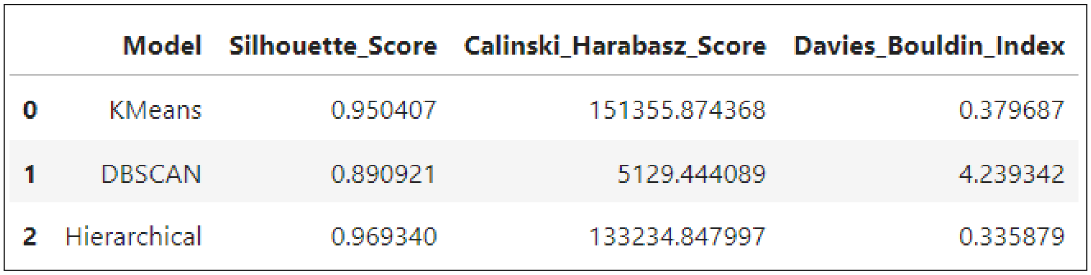
The DB index measures the average similarity ratio of each cluster with the cluster that is most similar to it. This involves both the within-cluster scatter (compactness) and the between-cluster separation (distance between clusters). This dual focus ensures that clusters are well-defined regardless of their shape. A lower DB index indicates better clustering. This means that clusters are more distinct from each other and the points within each cluster are closer to their centroid [24][4].

For the sample dataset, the Davies-Bouldin (DB) Index is computed, with results shown in Figures 21 and 22. In the ATU dataset, hierarchical clustering achieves the best score of 0.4557, followed by k-means clustering with 0.4838. DBSCAN performs the worst with a score of 1.2861. Conversely, in the LOTT dataset, hierarchical clustering also achieves the best score of 0.3358, followed by k-means with 0.3796. DBSCAN again has the worst performance, obtaining a score of 4.2393. These results highlight the varying effectiveness of different clustering algorithms across datasets.



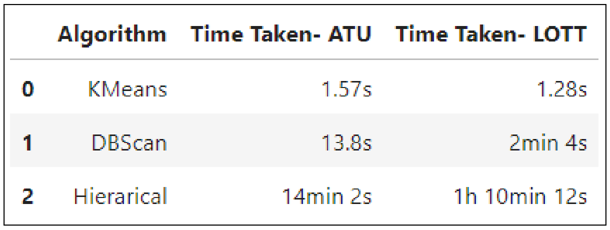
	Model	Silhouette_Score	Calinski_Harabasz_Score	Davies_Bouldin_Index
0	KMeans	0.968677	120218.148907	0.483887
1	DBSCAN	0.676930	1341.442045	1.286184
2	Hierarchical	0.989160	92184.610358	0.455711

Fig. 21. Table comparing 'ATU' evaluation result



	Model	Silhouette_Score	Calinski_Harabasz_Score	Davies_Bouldin_Index
0	KMeans	0.950407	151355.874368	0.379687
1	DBSCAN	0.890921	5129.444089	4.239342
2	Hierarchical	0.969340	133234.847997	0.335879

Fig. 22. Table comparing 'LOTT' evaluation result



	Algorithm	Time Taken- ATU	Time Taken- LOTT
0	KMeans	1.57s	1.28s
1	DBScan	13.8s	2min 4s
2	Hierarical	14min 2s	1h 10min 12s

Fig. 23. Execution time comparison of clustering algorithms across ATU and LOTT datasets

7 DISCUSSION

The research endeavours to explore and compare three distinct clustering approaches applied to a sales dataset. The primary objective is to analyse the outcomes of cluster analysis on sales data based on revenue and draw comparisons among them. As observed in Section 6, the evaluation of the applied clustering algorithm employs three metrics, to facilitate comparison of clustering results in this section. The evaluation of clustering results for the sample datasets, ATU (Figure 21) and LOTT (Figure 22), reveals interesting insights into the performance of different clustering algorithms.

In the case of 'ATU', k-means shows outstanding performance across all metrics. With a silhouette score of 0.9668, it indicates very well-defined clusters. The Calinski-Harabasz score for k-means is 120218, which is high, suggesting that the clusters are both compact implying the points are close to each other and well-separated indicating larger distances between cluster centroids. The Davies-Bouldin Index of 0.4838, showing low similarity between data points in different clusters (inter-cluster similarity) and thus good cluster separation. On the other hand, DBScan's performance on 'ATU' is notably lower. The Silhouette score of 0.6769 suggests that the clusters are not as well-defined as those generated by k-means. The CH score is lower at 1341, indicating less compact and more dispersed clusters. Furthermore, the DB Index of 1.2861 is quite high, reflecting poor separation between clusters. Hierarchical clustering on 'ATU' shows strong performance, with a silhouette score of 0.9891, which is the highest among the three methods. This score suggests extremely well-defined clusters. The CH score of 92184 is also high, although lower than that of k-means, indicating good clustering. The DB Index of 0.4557, being the lowest among the three methods, suggests excellent cluster separation.

For 'LOTT', K-means again demonstrates robust performance with a silhouette score of 0.9504, indicating well-defined clusters. Here robust refers to the stability and reliability of the clustering results. The CH score is high at 151355, showing very compact clusters with clear separation. The DB Index of 0.3796 is the lowest among the methods for 'LOTT', further confirming excellent cluster separation meaning high quality of cluster separation, as indicated by the low DB Index. DBSCAN, however, shows weaker performance on 'LOTT' overall. With a high silhouette score of 0.8909, indicating moderately defined clusters than other two applied methods meaning that the clusters formed are reasonably well-defined, but there is room for improvement. The CH Score of 5129 is lower among the methods for 'LOTT', indicating poorer cluster separation and internal cohesion. The DB Index of 4.2393 is high compared to all other results , suggesting poor separation and significant overlap between clusters. Hierarchical clustering on 'LOTT' performs very well, with a Silhouette Score of 0.9693, indicating well-defined clusters. The CH Score of 133234, while lower than that of K-Means, still shows good clustering performance. The DB Index of 0.3358 is the lowest, indicating the best separation among the clusters.

The results demonstrated in Figure 23 depicts execution time taken by applied algorithms. KMeans is the fastest on both datasets, reflecting its computational efficiency and scalability. DBScan took moderate amount of time to complete execution. While hierarchical clustering was the slowest, depiting its computational expensive behaviour. This indicates that KMeans is the most efficient algorithm, especially for larger datasets, while hierarchical clustering may become impractical for extensive data due to its high time complexity.

Overall, these results demonstrate the effectiveness of various clustering algorithms in managing large datasets. For both 'ATU' and 'LOTT', K-Means and Hierarchical clustering methods prove to be more effective, with K-means generally exhibiting a slight edge in overall clustering quality along with computation efficiency. Here effectiveness is referred to ability to produce high-quality clustering results. DBSCAN's lower performance suggests it may not be as suitable for large datasets, mainly due to its resource intensive nature and their density or distribution characteristics. These findings emphasize the importance of considering the inherent characteristics of the dataset and the clustering objectives when selecting an appropriate algorithm. While K-Means is often preferred for datasets with well-defined clusters, hierarchical clustering may be more suitable for data with hierarchical structures. DBSCAN's performance can vary depending on the presence of clear cluster structures within the dataset. Therefore, careful consideration of these factors is essential for achieving optimal clustering results in practice.

8 CONCLUSION

The study evaluated three clustering algorithm using various metrics on two datasets, ATU and LOTT. Based on the scores gained from three evaluation metrics and visual representation of results, K-means and hierarchical performed well, producing well-defined clusters validated by high scores. Conversely, DBScan did not perform well as per the scores gained by evaluation metrics. The research underscores the importance of using multiple evaluation metrics to assess clustering algorithms, providing a holistic understanding of their performance in terms of cluster quality, computational efficiency, and scalability. This multifaceted approach allows researchers and practitioners to make informed decisions about algorithm selection and parameter tuning, optimizing clustering outcomes in practical scenarios.

From the scatter plot results, it is evident that while clusters are separated by boundaries, they could be more distinctly divided with denser data points within each cluster. Some clusters display scattered data points, while others contain very few data points. The feature 'order quantity' does not significantly impact the clustering results. Conversely, 'price' shows a linear relationship with 'revenue,' indicating that both features equally influence the formation of clusters. K-Means clustering shows clear, well-separated clusters, especially in the lower range of the dataset. It effectively segments the data into distinct groups, making it easier to interpret. While some clusters might be forced into the data, especially if the number of clusters (k) is not optimal. The presence of outliers can be observed as clusters in the higher range have scattered data points. DBSCAN effectively identifies outliers, as seen from the scattered black dots labeled as outliers. It seems to cluster a significant portion of the data tightly around the origin, which is typical for this algorithm when handling dense clusters. But algorithm seems to struggle with identifying distinct clusters beyond the dense region near the origin. DBSCAN may not handle the high range of values in sample dataset well, leading to less distinct clusters. Hierarchical clustering appears to form clusters where maximum datapoints are part of one single cluster with fewer number of datapoints in other cluster.This might indicate that the algorithm is not entirely distinguishing between closely spaced data points. Hierarchical and dbscan clustering are computationally intensive for large datasets. For used sample datasets, K-Means seems to provide the most interpretable and distinct clusters. However, if outlier detection is critical, DBSCAN might be a useful supplementary method but by employing sampling dataset approach it can be more efficient for large datasets. Hierarchical clustering can provide a middle ground but may require fine-tuning and computational resources.

The primary motivation of this study was to cluster products based solely on revenue as the single primary feature. This focused approach provided clear insights into the clustering capabilities of different algorithms with respect to revenue-based segmentation. However, an interesting avenue for future research would be to incorporate additional features into the clustering process. By considering a more comprehensive set of product attributes, researchers could examine how the inclusion of these additional features influences the clustering outcomes. Such an investigation would reveal the extent to which the clusters vary and provide a deeper understanding of product segmentation. This expanded research could lead to more insights, enhancing the practical utility of clustering algorithms in diverse applications. Also exploring additional clustering algorithms and innovative methodologies could address the complexities inherent in diverse data landscapes. Furthermore, exploring emerging trends such as deep learning-based clustering or hybrid algorithms could offer novel solutions to complex clustering problems.

In conclusion, this study contributes to the advancement of clustering methodologies by providing an evaluation framework and insights into algorithm performance on sales dataset. By elucidating the strengths and limitations of various clustering algorithms, the research facilitates informed decision-making in algorithm selection and parameter optimization techniques. Ultimately, these efforts drive progress in clustering analysis, enabling more effective utilization of clustering techniques across various domains and applications.

REFERENCES

[1] 2015. State of Art of Different Clustering Approaches. *International Journal of Advanced Research in Computer and Communication Engineering* (2015), 81–89. https://doi.org/10.17148/IJARCCE.2015.4219

[2] 2017. Clustering retail products based on customer behaviour. 60 (2017), 752–762. https://doi.org/10.1016/J.ASOC.2017.02.004

[3] 2019. Towards better Validity: Dispersion based Clustering for Unsupervised Person Re-identification. *arXiv: Computer Vision and Pattern Recognition* (2019).

[4] 2022. Distance Analysis Measuring for Clustering using K-Means and Davies Bouldin Index Algorithm. *TEM Journal* (2022), 1871–1876. https://doi.org/10.18421/tem114-55

[5] Firas D. Ahmed, Aws Naser Jaber, Mohd Sharifuddin Ahmad, and Mazlina Binti Abdul Majid. 2015. Agent-based Big Data Analytics in retailing: A case study. In *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*. 67–72. https://doi.org/10.1109/ICSECS.2015.7333085

[6] Anil K. Jain and Richard C. Dubes. 1988. *Algorithms for clustering data.* Prentice-Hall, Inc., USA.

[7] A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data clustering: a review. *ACM Comput. Surv.* 31, 3 (sep 1999), 264–323. https://doi.org/10.1145/331499.331504

[8] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. 2002. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 7 (2002), 881–892. https://doi.org/10.1109/TPAMI.2002.1017616

[9] Tan Chun Kit and Nurulhuda Firdaus. 2021. Customer Profiling for Malaysia Online Retail Industry using K-Means Clustering and RM Model. *International Journal of Advanced Computer Science and Applications* (2021). https://api.semanticscholar.org/CorpusID:234322031

[10] Ahmed Abdullah Awadh Koofan and Mohammed Kaleem. 2020. Analyze and Enhance Sales in Lulu Supermarket using Data Mining Technology. *Journal of Student Research* (Jul. 2020). https://doi.org/10.47611/jsr.vi.926

[11] Robert Kwiatkowski. 2022. Customers clustering: K-Means, DBSCAN and AP. https://www.kaggle.com/code/datark1/customers-clustering-k-means-dbscan-and-ap

[12] Suzane Pereira Lima and Marcelo Dib Cruz. 2020. A genetic algorithm using Calinski-Harabasz index for automatic clustering problem. *Revista Brasileira de Computação Aplicada* 12, 3 (set. 2020), 97–106. https://doi.org/10.5335/rbca.v12i3.11117

[13] Chaoqun Ma and Jianhong Wu. 2007. *Data Clustering: Theory, Algorithms, and Applications*. Vol. 20. https://doi.org/10.1137/1.9780898718348

[14] Dhendra Marutho, Sunarna Hendra Handaka, Ekaprana Wijaya, and Muljono. 2018. The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News. In *2018 International Seminar on Application for Technology of Information and Communication*. 533–538. https://doi.org/10.1109/ISEMANTIC.2018.8549751

[15] NIKHIL. 2022. Setting up Evaluation Metrics. https://www.kaggle.com/code/nikhilkhetan/setting-up-evaluation-metrics#5.-Evaluation-metrics-for-clustering-(without-ground-truth-labels)

[16] Nashuha Omar, Nor Nazirun, Bhuwaneswaran Vijayam, Asnida Abdul Wahab, and Hana Bahuri. 2022. Diabetes subtypes classification for personalized health care: A review. *Artificial Intelligence Review* 56 (08 2022). https://doi.org/10.1007/s10462-022-10202-8

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[18] Komalpreet Kaur Pritika Talwar, Shubham. 2024. EXPLORING CLUSTERING TECHNIQUES IN MACHINE LEARNING. 12 (03 2024).

[19] Joerg Sander. 2010. *Density-Based Clustering*. Springer US, Boston, MA, 270–273. https://doi.org/10.1007/978-0-387-30164-8_211

[20] Ms Santhisree and Dr Damodaram. 2011. SSM-DBSCANand SSM-OPTICS : Incorporating a new similarity measure for Density based Clustering of Web usage data. *International Journal on Computer Science and Engineering* 3 (08 2011).

[21] Ketan Rajshekhar Shahapure and Charles Nicholas. 2020. Cluster Quality Analysis Using Silhouette Score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. 747–748. https://doi.org/10.1109/DSAA49011.2020.00096

[22] Ali Seyed Shirkhorshidi, Saeed Aghabozorgi, Teh Ying Wah, and Tutut Herawan. 2014. Big Data Clustering: A Review. In *Computational Science and Its Applications – ICCSA 2014*, Beniamino Murgante, Sanjay Misra, Ana Maria A. C. Rocha, Carmelo Torre, Jorge Gustavo Rocha, Maria Irene Falcão, David Taniar, Bernady O. Apduhan, and Osvaldo Gervasi (Eds.). Springer International Publishing, Cham, 707–720.

[23] Meshal Shutaywi and Nezamoddin N. Kachouie. 2021. Silhouette Analysis for Performance Evaluation in Machine Learning with Applications to Clustering. *Entropy* 23, 6 (2021). https://doi.org/10.3390/e23060759

[24] Akhilesh Kumar Singh, Shantanu Mittal, Prashant Malhotra, and Yash Vardhan Srivastava. 2020. Clustering Evaluation by Davies-Bouldin Index(DBI) in Cereal data using K-Means. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. 306–310. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00057

[25] Xu Wang and Yusheng Xu. 2019. An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index. *IOP Conference Series: Materials Science and Engineering* 569, 5 (jul 2019), 052024. https://doi.org/10.1088/1757-899X/569/5/052024