# A Performance Assessment on Unsupervised Machine Learning Models at GDP Datasets

**Rathindra Nath Mohalder*, Bijoy Podder, Mohammad Habibur Rahaman**

**Department of Computer Science and Engineering, Khulna Khan Bahadur Ahsanullah University, Bangladesh**

***Correspondence author**

**DOI: https://dx.doi.org/10.47772/IJRISS.2025.90600052**

**Received: 25 May 2025; Accepted: 29 May 2025; Published: 28 June 2025**

## ABSTRACT

Machine learning (ML) with unsupervised ML (UML) is a core engine for economic development across regions. UML enables novel solutions, improves efficiency, and boosts economic progress by letting advanced algorithms analyze large datasets with no labeled inputs. UML could have a substantial effect on the Gross Domestic Product. However, none of these speculative events might occur without overcoming severe challenges, including a shortage of skills to use it, ethical issues (see No Encoded Morality), data privacy problems, and alarmingly inequitable access to technology. By emphasizing UML development and adoption, countries will be able to move past these hurdles & they can wield their power, promoting a sustainable economy worldwide. Focusing on the unsupervised learning aspect, this work could further expand on how such breaks away from or augments supervised algorithms in similar applications. Using UML on GDP datasets helps to gain valuable insights and trends and facilitates data-driven decisions to enhance economic planning & policymaking. Some unique applications that are not possible with supervised approaches based on training from label data to input data, e.g., clustering, anomaly detection, and dimension reduction, have also been implemented by UML analyzing the GDP-related data without any label. UML requires no labeled data for training and is evaluated based on the performance of such models in extracting features, acting upon clusters, or reducing input space dimensions. UML Models are leveraged in multiple domains to gain insights from unlabeled data, make decisions, and optimize processes by using pattern discovery & data insights, anomaly detection, and flexibility across domains, further improving supervised learning. Unsupervised UML models find their way into an eclectic amalgamation of domains through which they reshape data-driven decision-making and operational workflows, sometimes even innovation. UML can transform, but it is not always used correctly. So, ethics and legality need to be accounted for, more so given social benefits through GDP.

**Keywords**: UML, DBSCAN, ML, BIRCH, GDP, EM, GMM.

## INTRODUCTION

Using UML on GDP data is a promising step in analyzing economic time series. GDP is complex due to its different kinds of data, including growth rates, sector contributions, terms like GNP and NNP, etc. UML presents a novel way of analyzing this data type to identify trends, group countries or regions with the same economic characteristics, and find anomalies that might threaten an unstable economy. This study aims to assess the comparative ability of different UMLs to improve economic analysis from GDP data. Dimensionality reduction (e.g., PCA, t-SNE) and clustering (e.g., K-means, DBSCAN) methods are the primary analysis points. Such models can consequently be used to find hidden economic relationships, cluster countries (or sectors) that behave similarly, and observe growth runaway behaviors against a baseline growth projection. The results of those assessments have far-reaching consequences. The implications of UML are significant; policymakers could use insights from UML for targeted interventions, international organizations would be able to provide a better market-leading economic prediction, and businesses would have greater insight into market dynamics. Nevertheless, for UML to be practical here, it must respond to the specific characteristics of GDP data: high dimensionality, sparsity, and interpretability [1]. A comparative analysis of the different algorithms on UML outlines their strengths, limitations, and implications for economic data-driven decision-making. It provides a deeper understanding of the role these methods play in economics.

## LITERATURE SURVEY

Gross Domestic Product (GDP) is one of the most critical measures of a country's health, combining production, income, and expenditure data. Third, GDP data can be highly complex and multivariate, given that different sectors, periods, and countries are involved. This creates a substantial analytical challenge, mainly when no labeled or explicit samples exist. This has led to increased attention towards UML methods, which are particularly suitable for discovering hidden patterns and associations in unlabeled data sets ·. Clustering algorithms (K-means, DBSCAN) and dimensionality reduction techniques such as PCA have shown promise for economic analysis on UML models. Clustering can cluster countries or regions that behave similarly financially, making it easier to compare them. At the same time, dimensionality reduction helps remove some dimensions of a large GDP dataset (the same as projecting from 3D space to 2D) and simplifies our analyses. They have helped discover economic irregularities, including sudden downturns and forecasting developments. More generally, UML is used across industries as examples for the segmentation of customers, detection of fraud, and subtyping diseases. In economics, it closes the gap in methods that are traditionally limited to small, fixed datasets by developing scalable approaches to NLP problems that can apply to oversized and dynamic datasets. The clustering of GDP data has also allowed the classification of countries based on their trade, and anomaly detection models are used for monitoring macroeconomic risks. UML methods, while helpful, can be challenged by hyperparameter sensitivity, interpretability challenges, and dependence on the domain expert for output validation. In addition, UML applications on economic datasets like GDP are still unexplored compared to health or finance areas. Such a study helps fill this gap by evaluating the performance of different UML models on GDP datasets, indicating their suitability and limitations for evidence-based, informed policy decision-making [2].

## METHODOLOGY

Machine learning algorithms can be mostly grouped as supervised and unsupervised learning depending upon the use of labels in the input data. The two major types of machine learning are supervised and unsupervised, where supervised uses labeled data, and the latter is based on unlabeled data. This paper presents explicitly the different classes of unsupervised learning in the context of disease prediction: partitioning clustering, model-based clustering, hierarchical clustering, and density-based clustering.

### Gaussian mixture Clustering:

This means that the Gaussian mixture model (GMM) is a parametric probabilistic model whereby each data point is assumed to be generated from a finite mixture of Gaussian distributions. These distributions fully describe the model. As such, it is a weighted sum of Gaussian components. All the information is captured in just three parameters for each Gaussian component: mean, covariance, and weight. These parameters are estimated from training data using the iterative expectation maximization (EM) algorithm [3]. The Gaussian Mixture Model is defined with the below probability density function:

$$P(x) = \sum_{k=1}^{k} \pi_k \cdot N(x|u_k, \Sigma_k) \text{ ……………………………………………….. (1)}$$

X = The data point, K: The number of Gaussian components (clusters), $\pi_k$: The mixing coefficient for the k-th Gaussian component ($\sum_{k=1}^{k} \pi_k = 1 \; and \; \pi_k \geq 0, N(x|u_k, \Sigma_k)$ : The probability density function of k-th Gaussian distribution is given by:

$$N(x|u_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}} \exp\left(\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right) \text{ ……….…………… (2)}$$

d is the dimensionality of the data, $\mu_k$ is the mean vector of the k-th Gaussian component, and Σk is the covariance matrix of the k-th Gaussian component (d×d).

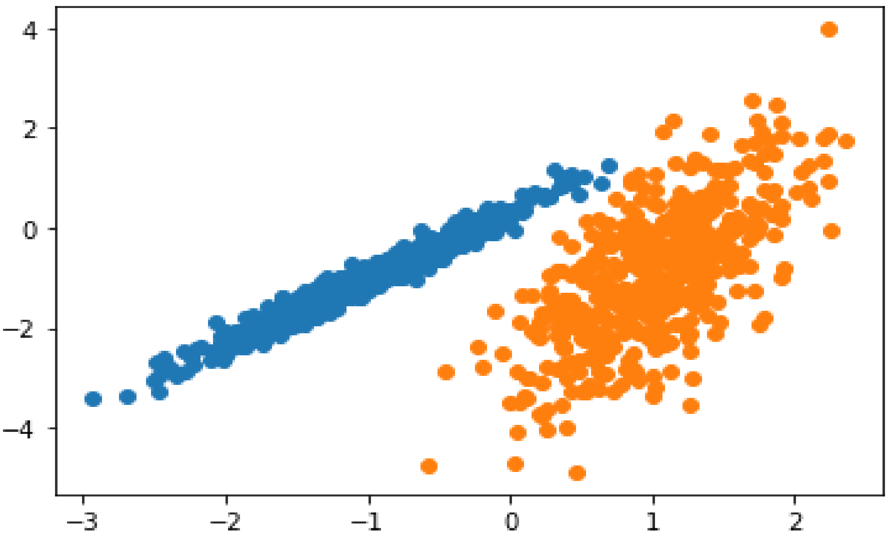

**Figure 01:** Gaussian mixture Clustering on GDP datasets

**Birch Clustering:**

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) is a scalable clustering algorithm for large datasets. It uses a hierarchical tree structure, the Clustering Feature Tree (CF Tree), to concisely describe the data. This summary allows BIRCH to cluster data incrementally and efficiently with limited memory resources, making it well-suited for dynamic or large-scale data. BIRCH minimizes the input-output (I/O) cost by working incrementally and can adapt to different cluster shapes, but it is best suited for data with convex-shaped clusters. It works in two primary steps:

1. Building a CF Tree from the dataset.
2. To refine the clusters, apply a clustering algorithm (e.g., k-means) on the tree's leaf entries.

Clustering Feature (CF):

A Clustering Feature summarizes a group of data points and is defined as a triple (N, LS, SS) where:

- N: The number of data points in the group.
- LS: The summation of all the data points $(LS = \sum_{i=1}^{N} x_i)$ ·
- SS: Sum Squared of Data Points $(SS = \Sigma_i^N = x_i^2)$

With these CF values, you can now calculate the following properties:

1. Centroid: C=LS/N
2. Radius (the compactness of the cluster):

Radius = $\sqrt{\frac{SS}{N} - \left(\frac{Ls}{N}\right)^2}$ ........................................................................ (3)

We can also measure the distance of points from each other to cluster Diameter (this is a coefficient of how widespread the points are within the specific cluster):

Diameter = $\sqrt{\frac{2}{N(N-1)}(N \cdot SS - LS^2)}$.......................................................... (4)

CF Tree Structure:

- Types of nodes: Root (Internal) Errors Nodes and leaves.
- Threshold (T): The maximum distance of a cluster that can exist in a node. It's threshold for the level of granularity that you want to cluster

Algorithm Workflow:

Phase 1: Construction of the CF Tree:

- Data is scanned progressively, and each point is inserted in the CF Tree.
- If the point falls into an existing cluster (T is the threshold of similarity or distance), the CF of that cluster is updated. If not, then create a new cluster.

Phase 2: Global Clustering:

- The final clustering uses a standard clustering algorithm (k-means or agglomerative clustering), which inputs the CF Tree leaf entries.

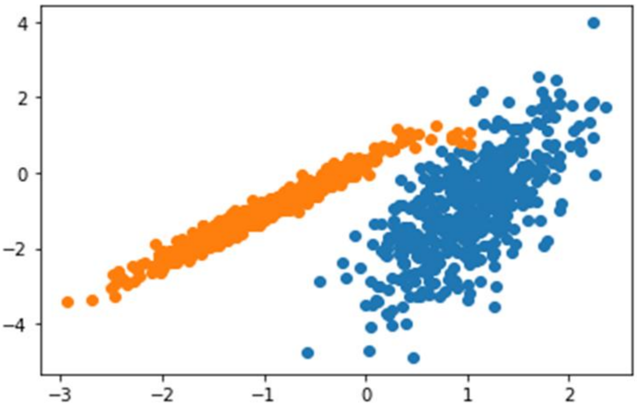

**Figure 02:** BIRCH Clustering on GDP datasets

**Spectral Clustering:**

More advanced are methods that work with the graph representation of the set, such as spectral clustering — a technique that is particularly good for nonconvex distributions. The process treats the data as a graph; each data point is treated as a node in the semantic space, and edges represent similarities or distances between nodes. Using the eigenvalues and eigenvectors of the graph Laplacian matrix, the data is embedded in a lower-dimensional space where we can apply some classical clustering methods such as k-means or Gaussian Mixture Models.

**Steps and Key Equations:**

1. Create the Affinity Matrix (W): It evaluates pairwise similarity among each data point using the kernel (e.g., gaussian kernel).

2. Graph Laplacian (L):

   Unnormalized Laplacian: $L=D-W$ ………………………………………………….. (5)

   Or normalized variants:

   $L_{sym}=D^{-1/2}LD^{-1/2}$ or $L_{rw}=D^{-1}L$, ……………………………………………………. (6)

Where D is the degree matrix: $Di = \Sigma_j w_{ij}$

3. Eigendecomposition: Calculate the k smallest eigenvectors of L, which gives the embedding matrix X
4. Clustering: Run a clustering algorithm on the rows of X in transformed space.

The clusters are the actual partitioning of the graph nodes, basically putting together these data points [5].

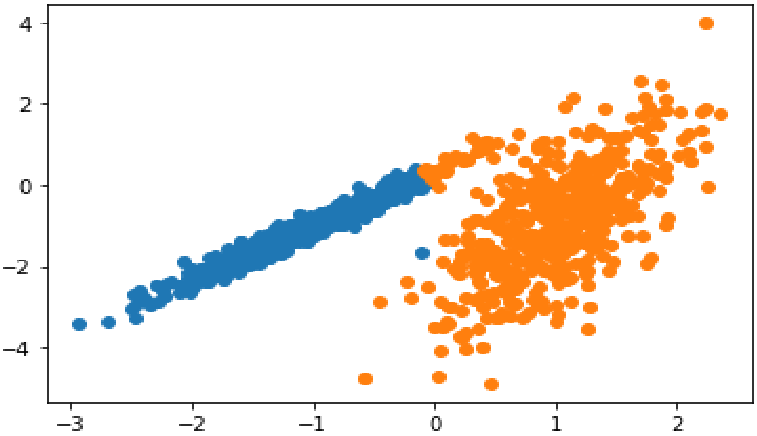

**Figure 03:** Spectral Clustering on GDP datasets

**Dbscan Clustering:**

Density-based spatial clustering of applications with noise (DBSCAN) is a well-known density-based clustering algorithm. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) can find clusters of arbitrary shapes and sizes. It has a particularly effective way of detecting outliers by identifying clusters based on high-point density regions separated by low-density regions.

- **Core Point**: A point with at least min Pts neighbors within a radius ε.
- **Border Point**: A point that is not a core point but lies within ε of a core point.
- **Noise Point**: A point neither a core nor a border point.

Compute the distance matrix or use a distance function to find neighborhoods. Identify core points based on ε neighborhood density:

$$N(p) = \{q \in D | dist(p,q) \leq \varepsilon\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

Where N(p) is the neighborhood of point p and dist is a distance function. Expand clusters iteratively, starting from core points and including all density-reachable points.

DBSCAN is popular across many domains, including spatial data analysis, anomaly detection, and clustering of biological data. Strategies in recent advances concentrate on improving computational efficiency and adapting to high dimensional or streaming data scenarios [6][7][8].

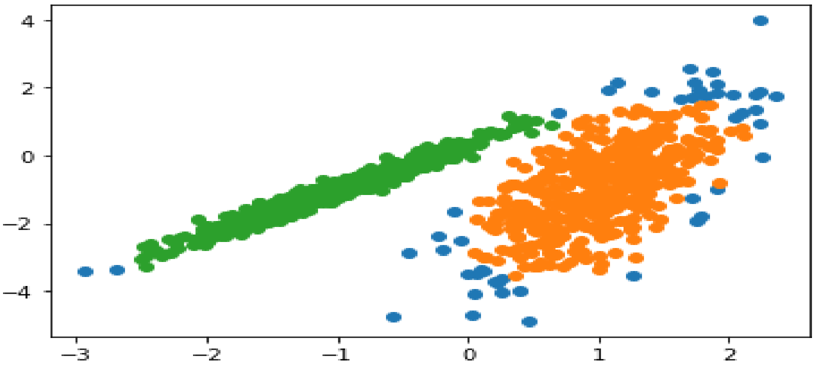

**Figure 04:** DBSCAN Clustering on GDP datasets

### Mean-Shift Clustering:

Mean-shift clustering is a non-parametric, density-based clustering method that finds a density function's modes (peaks). DBSCAN can be considered iterative in moving data points towards midpoints of higher density, so it does not necessitate the specification of several clusters beforehand and is relatively efficient at finding arbitrary-shaped clusters.

- **Kernel Density Estimation (KDE)**: This algorithm implements a kernel function (Gaussian Kernel) for estimating the density surrounding each data point.

- **Mean Shift Update**: At each iteration, the data points are updated using:

$$m(x) = \frac{\Sigma_{x_i \in N(x)} k(x_i - x) x_i}{\Sigma_{x_i \in N(x)} k(x_i - x)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

Where m(x) is the mean-shift vector, K is the kernel function, and N(x) represents the neighborhood within a bandwidth h. This formula pulls points toward the nearest mode.

- **Convergence**: The process continues until convergence and similar points get clustered around their respective modes.

- **Output**: Clusters are defined using modes outputted depending on data, so points are allocated to different clusters according to the closest mode (i.e., center). Mean shift is often used in image processing, such as computer vision and bioinformatics, whenever we have irregular shape clusters. Recent enhancements, such as the robust mean-shift state-of-the-art method, are expected to manage noise better and achieve faster convergence speed [9][10].

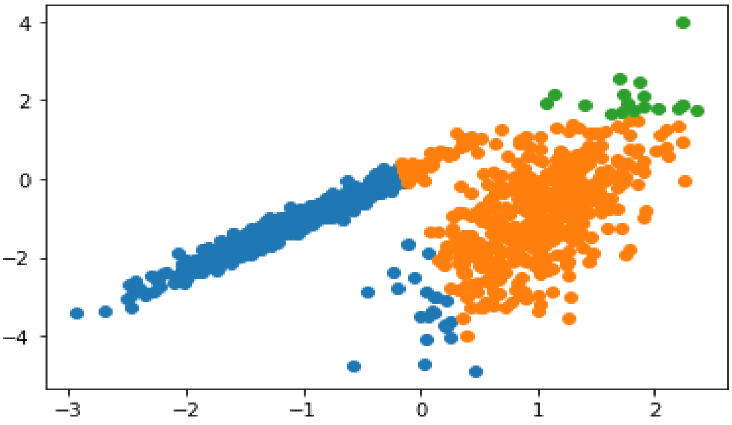

**Figure 05:** Mean-Shift Clustering on GDP datasets

### Agglomerative Clustering:

Agglomerative clustering is a type of hierarchical clustering that merges clusters sequentially. Starting with each data point as a separate cluster, it iteratively merges clusters according to a selected linkage criterion (minimum, maximum, and average distance). The procedure is repeated until all data points reside in one cluster or the number of clusters reaches the desired number.

1. **Pairwise Distance Calculation**:

Calculate pairwise distances d(xi,xj) between data points using a distance function; for example, one could use Euclidean or Manhattan distance: $d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_j)^2}$ ................................ (9)

2. **Linkage Criteria:** Let d between two clusters Ci and Cj shall be defined as:
   - **Single Linkage (Minimum Distance)**: $d(C_i, C_j) = \min_{x \in c_i y \in c_j} d(x,y)$ ………………….. (10)
   - **Complete Linkage (Maximum Distance)**: $d(C_i, C_j) = \max_{x \in c_i y \in c_j} d(x,y)$…………….. (11)
   - **Average Linkage (Mean Distance)**: $d(C_i, C_j) = \frac{1}{|c_i||c_j|} \sum_{x \in c} \sum_{x \in c} d(x, y)$ ……………………… (12)
3. **Cluster Merging**: Iteratively join the two closest clusters based on a given similarity measure (linkage criterion)
4. **Dendrogram Construction**: Create a dendrogram to visualize the cluster hierarchy.

Agglomerative clustering has been extensively applied to problems in Bioinformatics, Document clustering for NLP, and social network analysis [11].

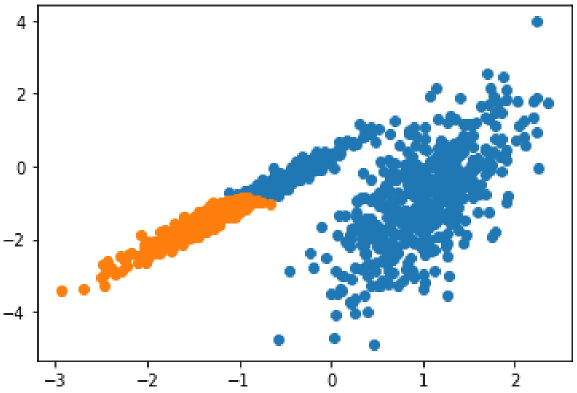

**Figure 06:** Agglomerative Clustering on GDP datasets

### K-Means Clustering:

K-Means Clustering: So far, we have learned about primary clustering and supervised learning for the different flavors of K-means. Chosen a data set and create Clusters- each cluster holding some points. It minimizes the within-cluster variance, iteratively refining cluster centers (centroids) and reassigning data points. The goal is to group data points such that the total intra-cluster variance is minimized, producing compact and distinct clusters. The algorithm minimizes the sum of squared distances between data points and their assigned cluster centroids:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - u_i\|^2 \text{ ……………………………………………………………….. (13)}$$

Here:

- $C_i$ is the set of points in cluster i,
- $\mu_i$ is the centroid of cluster I,
- $\|x-\mu i\|^2$ represents the Euclidean distance between a point xxx and its cluster centroid.

K-Means has been updated and used in many aspects. However, it still faces difficulties regarding initialization sensitivity, scalability issues, and low capabilities to work with high-dimensional or imbalanced datasets. K-Means++) Better initialization of centroids Incorporating new distance metrics Parallel implementations olefine Adopting for extreme clusters tests to strengthen performance [12][13].

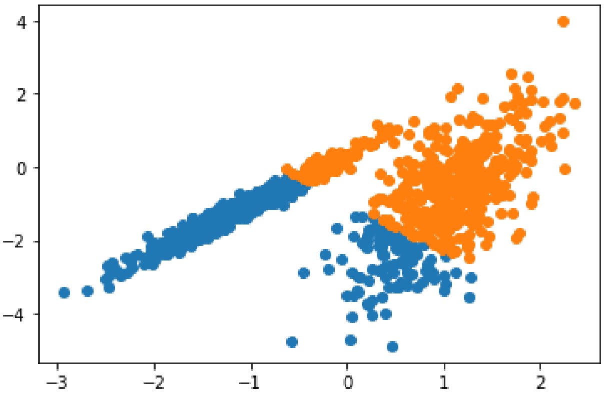

**Figure 07:** K-Means Clustering on GDP datasets

**Affinity Propagation Clustering:**

Affinity Propagation (AP) is a clustering algorithm that identifies exemplars—representative data points for each cluster—through message passing between data points. Unlike algorithms like K-Means, it does not require the number of clusters to be pre-specified and works on pairwise similarity measures between points [14].

**Similarity Matrix** (S):

Define the similarity between data points i and j as:

$S(i,j)=-\text{distance}(x_i,x_j)$ …………………………………………… (14)

where higher values indicate greater similarity.

**Responsibility and Availability Updates**:

The algorithm uses two matrices:

- **Responsibility (R(i,k))**: Measures the suitability of point k to be the exemplar for i:

$R(i,k) \leftarrow S(i,k) - \max_{k \neq k'}(A(i,k') + S(i,k'))$ ……………………… (15)

- **Availability (A(i,k))**: Reflects the evidence for point i selecting k as its exemplar:

$A(i,k) \leftarrow \min\ (0, R\ (k,k + \sum_{i' \notin \{i,k\}} \max\ (0, R(i',k)))$ …………… (16)

**Convergence**: Iteratively update R(i,k) and A(i,k) until clusters stabilize.

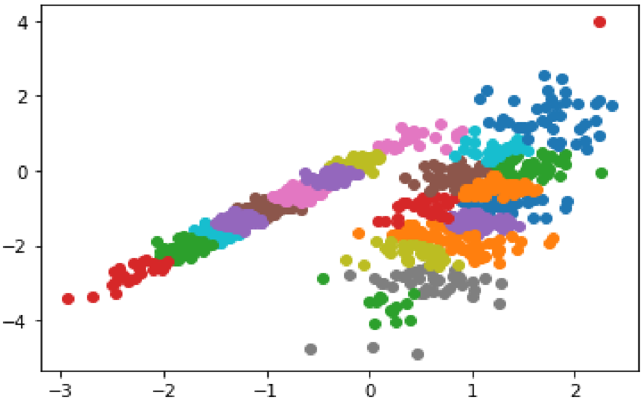

**Figure 08:** Affinity Propagation Clustering on GDP Datasets

**OPTICS Clustering:**

OPTICS is a density-based clustering that extends DBSCAN (Density-Based Spatial Clustering of Applications with Noise). DBSCAN is noise-sensitive, and traditional density-based clustering methods use a fixed density threshold. This ordering helps extract clusters at different density levels without explicitly requiring cluster parameters upfront [15][16].

**Reachability Distance**:

The reachability distance of a point p from another point o is defined as:

Reachability-Dist (p,o)=max(ϵ,Dist (p,o)) ………………………….. (17)

Where ϵ is a neighborhood radius parameter, and Dist (p,o) is the distance between points p and o.

**Core Distance**:

The core distance of a point o is the smallest distance such that at least MinPts points are within its ϵ-neighborhood:

Core-Dist (o)=min (Dist (o,p)) for p satisfying |Nϵ(o)| ≥ MinPts …….(18)

**Cluster Extraction:**

OPTICS uses a priority queue to process points in increasing order of their reachability distance, constructing a reachability plot that helps extract clusters by visualizing valleys in reachability values.

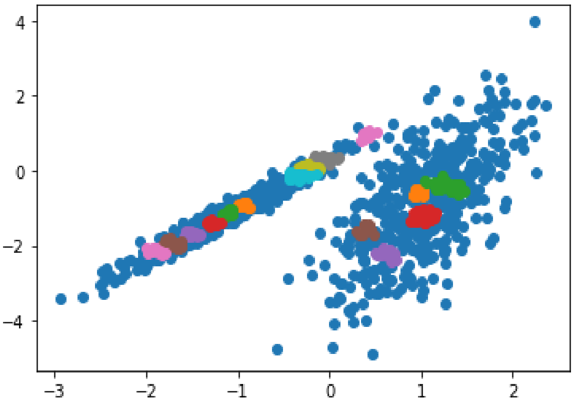

**Figure 09:** OPTICS Clustering on GDP Datasets

## EXPERIMENTATION AND RESULT ANALYSIS

**Dataset:**

The details of datasets used in this study 167 Table 1 were displayed, with ten columns. Here is a table showing the ten columns where we use data types. To ensure that no data pre-processing has influenced detection, this research uses the data in its raw form. We drop any entries with missing values.

**Table 1**: Column description

| Column Name | Description |
|---|---|
| country | Name of the country |
| child_mort | Death of children under 5 years of age per 1000 live births |

| | |
|---|---|
| exports | Exports of goods and services per capita. Given as %age of the GDP per capita |
| health | Total health spending per capita. Given as %age of GDP per capita |
| imports | Imports of goods and services per capita. Given as %age of the GDP per capita |
| Income | Net income per person |
| Inflation | The measurement of the annual growth rate of the Total GDP |
| life_expec | The average number of years a newborn child would live if the current mortality patterns remained the same. |
| total_fer | total Fertility Rate – the number of children each woman would have if the current age-fertility rates remain the same. |
| gdpp | The GDP per capita. It is computed as the Total GDP divided by people. |

**Result Analysis**

**Top Performers:**

- Surprisingly, Gaussian Mixture Clustering achieves the highest accuracy by 96.24%, revealing its ability to model the given data's hidden structure properly.
- BIRCH Clustering: 93.65%; BIRCH Clustering also performs at a winning level. This can be expected since this algorithm has efficient procedures, which work great, especially if the data is large enough and hierarchically structured.

**Mid-Range Performers:**

- Performance ranges moderately for Spectral Clustering (79.48%) and DBSCAN (77.42%). Though it can detect non-linear and arbitrarily shaped clusters, clustering accuracy may degrade on high noise and varying density data.

**Lower Performers:**

- Mean-shift clustering (58.15%) performs rather poorly, which may be due to its sensitivity to bandwidth selection.
- Finally, in 6th place, Agglomerative Clustering (44.62%), followed by K-Means Clustering (25.29%) and Affinity Propagation (22.00%), performs the worst, suggesting that it struggles to deal with complex cluster structures or high-dimensional datasets.
- Even though OPTICS (15.01%) can detect clusters of various densities, it also fails, which may indicate issues in the data, such as high noise or tightly packed clusters.

**Table 2**: Represent the nine clustering algorithms' accuracy

| **NO.** | **Clustering Algorithm Name** | **Accuracy** |
|---|---|---|
| 1. | Gaussian mixture Clustering | 96.24% |
| 2. | BIRCH Clustering | 93.65% |
| 3. | Spectral Clustering | 79.48% |
| 4. | DBSCAN Clustering | 77.42% |
| 5. | Mean-Shift Clustering | 58.15% |
| 6. | Agglomerative Clustering | 44.62% |
| 7. | K-Means Clustering | 25.29% |

| | | |
|---|---|---|
| 8. | Affinity Propagation Clustering | 22.00% |
| 9. | OPTICS Clustering | 15.01% |

Figure 10 represents the clustering algorithms' accuracy. These two algorithms provide consistent performance, making them suitable for datasets with well-defined or hierarchical structures. The variability in accuracy reflects the impact of dataset characteristics (e.g., noise, cluster density, shape). Although DBSCAN and spectral Clustering work well when the shape is arbitrary, they have issues handling different densities and overlapping clusters. Because its mathematical foundation is an oversimplistic assumption of spherical clusters, K-Means is not good with complicated data and is highly inaccurate. OPTICS has a solid theoretical background for varied density cluster detection; however, its implementation may not be robust on this dataset, leading to its lowest accuracy.

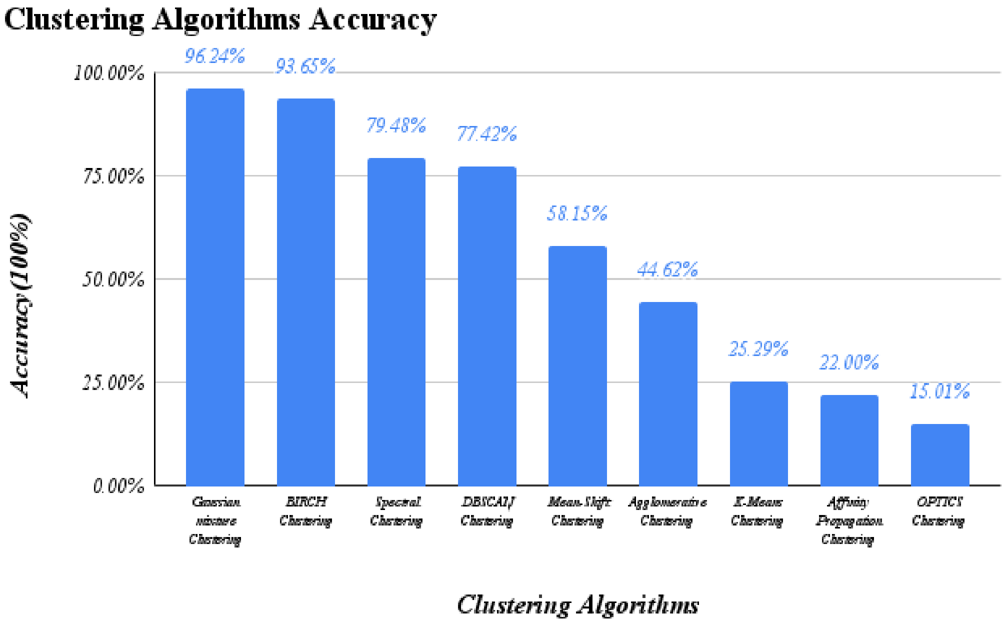

**Figure 10:** represents the clustering algorithms' accuracy

## DISCUSSION

These results answer the question of the characteristics of datasets — distribution, noise, density variations, and cluster shapes play essential roles in the performance comparison between different clustering algorithms. When the data is dense, Algorithms with Gaussian assumptions (e.g., Gaussian Mixture) perform better than those with complex cluster assumptions (e.g., K-Means). More flexible algorithms like DBSCAN and Spectral Clustering that deal with arbitrary shapes and densities perform better than rigid methods such as K-Means. On the other hand, if parameters (e.g., DBSCAN) are not optimized, accuracy can be reduced due to its sensitivity. Scalability and efficiency make high-performing algorithms like BIRCH optimal for large datasets. On the other hand, Spectral Clustering is a computationally expensive method and may not be fit for high-dimensional data or larger datasets. Clustering is highly dependent on the dataset and clustering goals, and this analysis underlines how important it is to compare clustering algorithms in context. Clustering is widely used in research and practical applications, but its effectiveness relies on proper algorithm choice, parameter tuning, and data preprocessing.

## CONCLUSION AND FUTURE WORK

The work shows that different unsupervised learning models (based on clustering methods or dimensionality reduction techniques) achieve variable quality outcomes regarding GDP datasets. Gaussian Mixture Models (GMM) and BIRCH are popular models with high accuracy among the best-performing clustering algorithms since they can efficiently process multivariate and large-scale data. Model Performance Variance: The observed differences in model performance are primarily due to the intrinsic nature of GDP data, noise, and the dimensionality involved. DBSCAN, a density-based method, fails to cluster the high dimensional GDP because it is sensitive to the parameters. The results imply that the machine learning model should be selected based on an analytical goal (anomaly detection, regional clustering, or economic forecasting). Furthermore, interpretable models, like hierarchical clustering, could be more appropriate in a policymaking setting. This paper highlights challenges in the field, such as high-complexity computation time, parameter sensitivity, and difficulties with validating unsupervised results based on known ground truth. Such challenges emphasize the importance of preprocessing, parameter tuning, and domain-specific validation schemes. The work further suggests investigating hybrid methods capitalizing on the benefits from the diverse blending of models (e.g., clustering + dimensionality reduction) to perform a better-compromising performance between classification and reconstruction levels. The other point it suggests is to infuse/ embed domain knowledge, which will help better model interpretation and align results with the real-world application. These results align with some general conclusions from machine learning and GDP forecasting literature, indicating that Random Forests and XG-Boost work well as candidate models on more structured and mixed datasets. On the other hand, more straightforward statistical modeling approaches such as ARIMA are still competitive for certain specific tasks.

### Author Contributions

J.M.D.—Conceptualization, methodology, validation, formal analysis, investigation, resources, writing—review and editing, supervision, project administration, funding acquisition. M.V.—software, formal analysis, data curation, writing—original draft preparation. All authors have read and agreed to the published version of the manuscript.

### Funding

This research received no external funding.

### Conflict of Interest Statement

The authors declare no conflicts of interest.

### Data Availability Statement

The data and supporting results will be provided on request due to privacy and confidentiality concerns.

## REFERENCE

1. Haohui Lu, Shahadat Uddin, “Unsupervised machine learning for disease prediction: a comparative performance analysis using multiple datasets,” Health and Technology (2024) 14:141–154,https://doi.org/10.1007/s12553-023-00805-8
2. Tim Callen, “Gross Domestic Product: An Economy’s All,” finance & development, https://www.imf.org/en/Publications/fandd/issues/Series/Back-to-Basics/gross-domestic-product-GDP.
3. Stefano Michieletto, Francesca Stival, Enrico Pagello, “Chapter 3 - A probabilistic approach to reconfigurable interactive manufacturing and coil winding for Industry 4.0”, Advances in Mathematics for Industry 4.0, 2021, Pages 61-93
4. Tian Zhang, Raghu Ramakrishnan, Miron Livny, “BIRCH: an efficient data clustering method for extensive databases” In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (pp. 103-114). ACM. DOI: 10.1145/233269.233324
5. Cinzia Di Nuzzo, "Advancing Spectral Clustering for Categorical and Mixed-Type Data: Insights and Applications," Multidisciplinary Digital Publishing Institute (MDPI)mathematics journal, 2024, 12, 508. https://doi.org/10.3390/math1204050
6. Omkaresh Kulkarni, Adnan Burhanpurwala, "A Survey of Advancements in DBSCAN Clustering Algorithms for Big Data," 3rd International Conference on Power Electronics and IoT Applications in Renewable Energy and its Control (PARC), 2024, DOI:10.1109/PARC59193.2024.10486339.

7. R. Dhivya, N. Shanmugapriya, "An Efficient DBSCAN with Enhanced Agglomerative Clustering Algorithm," 4th International Conference on Electronics and Sustainable Communication Systems (ICESC), Published:2023, DOI: 10.1109/ICESC57686.2023.10193224.
8. Daren Wang, Xinyang Lu, Alessandro Rinaldo, "DBSCAN: Optimal Rates for Density-Based Cluster Estimation," Journal of Machine Learning Research 20 (2019) 1-50, https://jmlr.org/papers/volume20/18-470/18-470.pdf
9. Claude Cariou; Steven Le Moan; Kacem Chehdi, "A Novel Mean-Shift Algorithm for Data Clustering," Published in IEEE Access, January 31, 2022, Digital Object Identifier 10.1109/ACCESS.2022.3147951
10. Itshak Lapidot, “Stochastic mean-shift clustering,” submitted to Elsevier, December 27, 2023, https://doi.org/10.48550/arXiv.2312.15684
11. Emamjomeh-Zadeh et al., “Fair Algorithms for Hierarchical Agglomerative Clustering,” IEEE Transactions on Knowledge and Data Engineering, 2023, DOI: 10.1109/TKDE.2023.
12. Mohiuddin Ahmed, Raihan Seraj and Syed Mohammed Shamsul Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation," Multidisciplinary Digital Publishing Institute (MDPI) electronics journal, Electronics 2020, 9, 1295; doi:10.3390/electronics9081295
13. Belal Abuhaija, Namig Isazade, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," April 2023, Information Sciences, Vol. 622, pp 178-210, doi.org/10.1016/j.ins.2022.11.139
14. [Michele Leone, Sumedha, Martin Weigt, "Article Navigation Journal Article Clustering by soft-constraint affinity propagation: applications to gene-expression data," Bioinformatics, Volume 23, Issue 20, October 2007, Pages 2708–2715, https://doi.org/10.1093/bioinformatics/btm414
15. Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jörg Sander, “OPTICS: Ordering Points to Identify the Clustering Structure”, Proc. ACM SIGMOD’99 Int. Conf. on Management of Data, Philadelphia PA, 1999.
16. Jing Wang, Daniel K Schreiber, Nathan Bailey, Peter Hosemann, Mychailo B Toloczko, "The Application of the OPTICS Algorithm to Cluster Analysis in Atom Probe Tomography Data," Microscopy and Microanalysis, Volume 25, Issue 2, 1 April 2019, Pages 338–348, https://doi.org/10.1017/S1431927618015386