**Gaussian Mixture Models (GMM) vs. Traditional Clustering Techniques**

*Binal Patel, Kaleem Mohammed, Siddhi Naik*

## 1. Introduction

### Overview of GMM and Its Significance

Gaussian Mixture Models (GMM) is a probabilistic clustering technique that models data as a mixture of multiple Gaussian distributions. Unlike k-Means, which assigns each data point to a single cluster, GMM allows soft clustering, meaning that each point belongs to multiple clusters with certain probabilities. This flexibility makes GMM effective in scenarios where data exhibits overlapping clusters or complex structures.

### Relevance to Machine Learning

GMM is widely used in machine learning due to its ability to model complex data distributions. Unlike traditional clustering methods like k-Means, which assumes spherical clusters, GMM provides a more robust approach by allowing clusters to have different shapes and sizes. The expectation-maximization (EM) algorithm used in GMM ensures that it iteratively improves cluster assignments, leading to more accurate clustering results.

### Comparison with Traditional Algorithms

* **k-Means**: Uses hard clustering; each data point is assigned to only one cluster.
* **Hierarchical Clustering**: Builds a hierarchy of clusters but lacks probabilistic assignments.
* **DBSCAN**: Density-based clustering that detects noise but struggles with varying densities.
* **GMM**: Assigns probabilities to clusters, enabling a more flexible clustering approach.

### Choice of Datasets

The datasets selected for this study were chosen to evaluate clustering models in different scenarios:

* **Moons Dataset**: Features two crescent-shaped clusters, making it a good test for non-spherical clustering.
* **Blobs Dataset**: Contains well-separated spherical clusters, providing an ideal scenario for k-Means and Hierarchical Clustering.
* **Circles Dataset**: Has nested circular clusters, testing how well models handle concentric patterns.

## 2. How It Works

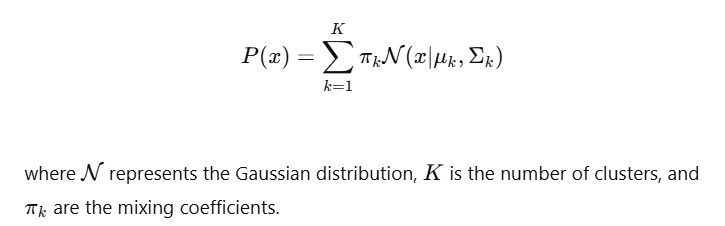
### Fundamental Principles

GMM assumes that data is generated from a mixture of several Gaussian distributions. Each cluster is represented by three parameters:

1. **Mean (μ)**: The center of the Gaussian distribution.
2. **Covariance (Σ)**: The spread or shape of the distribution.
3. **Mixing Coefficients (π)**: The proportion of points belonging to each Gaussian component.

### Mathematical Formulation

The probability density function (PDF) for a Gaussian mixture is given by:



### Expectation-Maximization (EM) Algorithm

GMM is trained using the **EM algorithm**, which consists of two steps:

* **Expectation Step (E-Step)**: Compute the probability of each data point belonging to each Gaussian component.
* **Maximization Step (M-Step)**: Update the parameters (means, covariances, and weights) to maximize the likelihood of the observed data.

### Key Parameters

* **n\_components**: Number of Gaussian distributions.
* **covariance\_type**: Specifies how covariance is calculated (full, tied, diagonal, spherical).
* **tol**: Convergence threshold.
* **max\_iter**: Number of iterations for the EM algorithm.

## 3. Use Cases & Performance

### Where GMM Works Best

* When data has overlapping clusters (e.g., Moons Dataset).
* In cases where clusters have different shapes and variances.
* When probabilistic clustering is needed.

### Advantages and Limitations

|  |  |  |
| --- | --- | --- |
| *Feature* | *Advantages* | *Limitations* |
| *Cluster Shape* | Works with elliptical clusters | Assumes data follows a Gaussian distribution |
| *Soft Clustering* | Assigns probabilities | Sensitive to initialization |
| *Flexibility* | Handles overlapping clusters | Computationally expensive |

### Real-World Applications

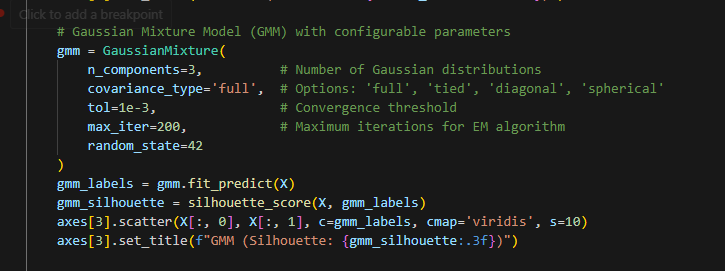
* **Medical Imaging**: Clustering different tissue types in MRI scans.
* **Marketing**: Customer segmentation based on spending behavior.

## 4. Code Implementation

### Python Implementation of GMM

Full implementation available on GitHub**:** [Machine Learning Capstone Project - main.ipynb](https://github.com/Kaleem-QADR/Machine_Learning_Capstone_Project/blob/main/scripts/main.ipynb)

Below is a snippet from the main implementation used in our analysis:



### ****Key Parameters****

The following parameters play a crucial role in controlling the behavior and performance of the GMM algorithm:

* **n\_components (Number of Gaussian Distributions):** Defines the number of Gaussian components (clusters) to be fitted. Choosing an appropriate value impacts how well the model captures the underlying structure of the data. In our study, we determined optimal values using **BIC & AIC selection**, ensuring that GMM does not overfit or underfit.
* **covariance\_type (Covariance Calculation Method):** Determines how the covariance matrix is handled for each Gaussian component. We used:
  + 'full' (each component has its own covariance matrix) for **maximum flexibility** in cluster shape.
  + 'diagonal' and 'spherical' for **simplified assumptions** in some datasets where we expect more uniform distributions.
* **tol (Convergence Threshold):** Specifies the stopping criteria for the Expectation-Maximization (EM) algorithm. A lower tol value (e.g., 1e-3) ensures finer convergence but increases computation time, while a higher value might lead to early stopping with suboptimal results.
* **max\_iter (Maximum Iterations for EM Algorithm):** Defines how many iterations EM can run before terminating. We experimented with values such as 200-300 to balance accuracy and efficiency, ensuring that the model reaches convergence without excessive computation.

### Evaluation Tests and Their Relevance

* **Silhouette Score**: Measures the separation and compactness of clusters (applies to all models). Ranges from **-1 to 1**; higher values indicate well-defined clusters.
* **BIC & AIC**: Used **only for GMM**, since it is a probabilistic model, unlike k-Means, DBSCAN, and Hierarchical Clustering.

## 5. Comparison & Visualization

### Comparison with Traditional Clustering Methods

**Silhouette Scores for each model on the three dataset**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Moons Dataset** | **Blobs Dataset** | **Circles Dataset** |
| **k-Means** | **0.441** | **0.787** | **0.381** |
| **Hierarchical** | 0.381 | 0.783 | 0.351 |
| **DBSCAN** | 0.386 | 0.720 | 0.004 |
| **GMM** | 0.435 | 0.783 | **0.381** |

### BIC & AIC Scores for GMM Model Selection

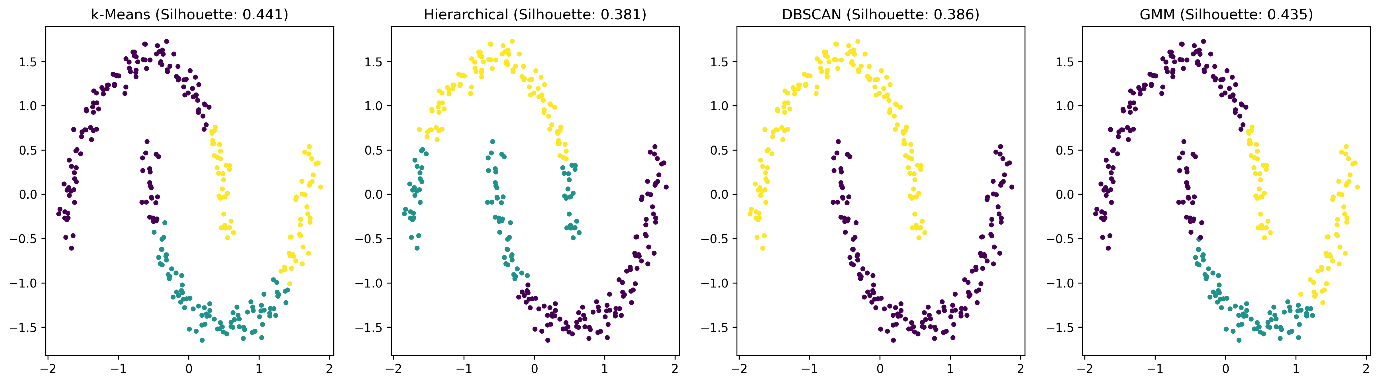
|  |  |  |
| --- | --- | --- |
| *Dataset* | *Optimal Components (BIC)* | *Optimal Components (AIC)* |
| *Moons* | 8 | 9 |
| *Blobs* | **3** | 6 |
| *Circles* | 9 | 9 |

### Visualization and Interpretation

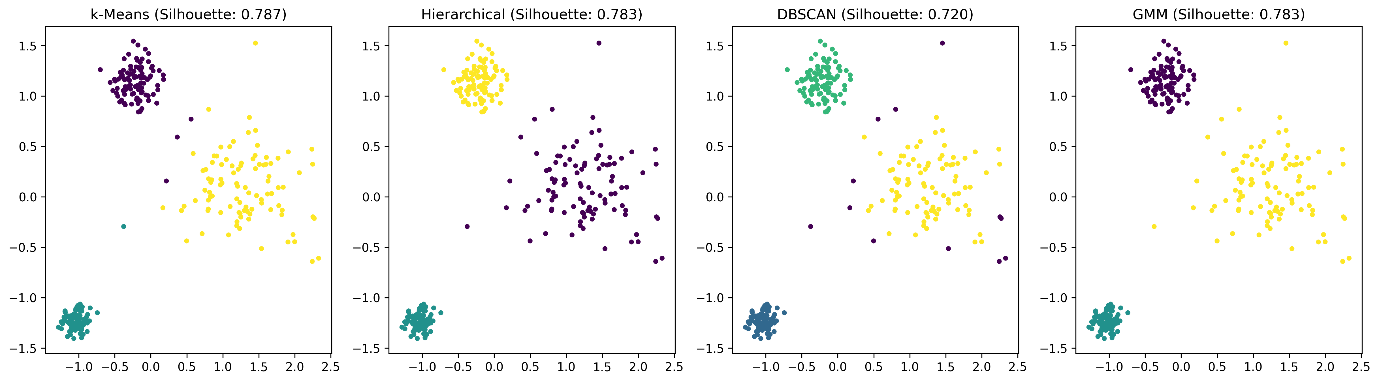
The following figures illustrate clustering results and evaluation metrics:

* **Fig. 1:** Scatter plots of clusters formed by different models.
* **Fig. 2:** BIC & AIC Graphs for GMM model selection.
* **Fig. 3:** Cluster Probability Histogram showcasing the soft clustering effect in GMM.

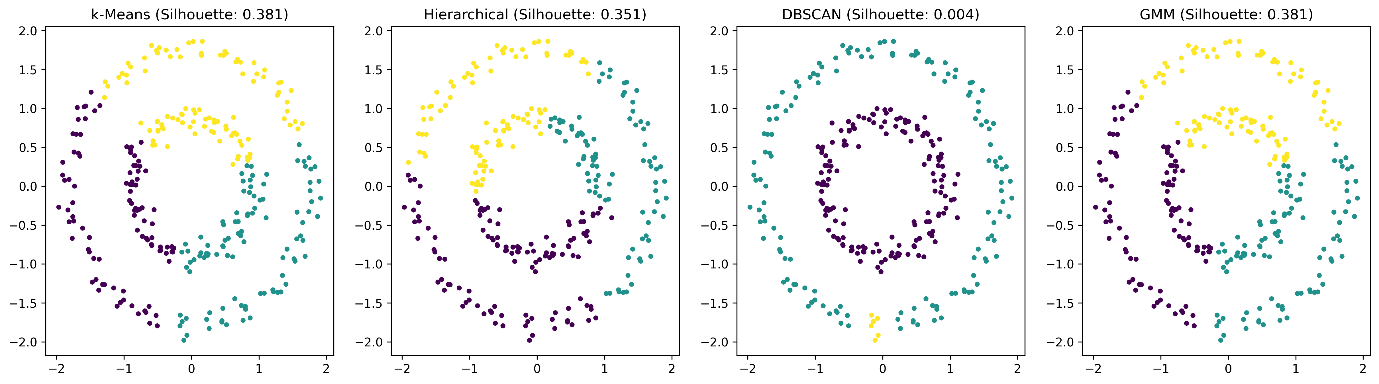
**Fig. 1: Clustering Results:** *Scatter plots showing the clustering assignments for each dataset using different models*

****

The **Moons Dataset** reveals that **DBSCAN and GMM** perform better than k-Means, as they can capture non-linear structures.

**

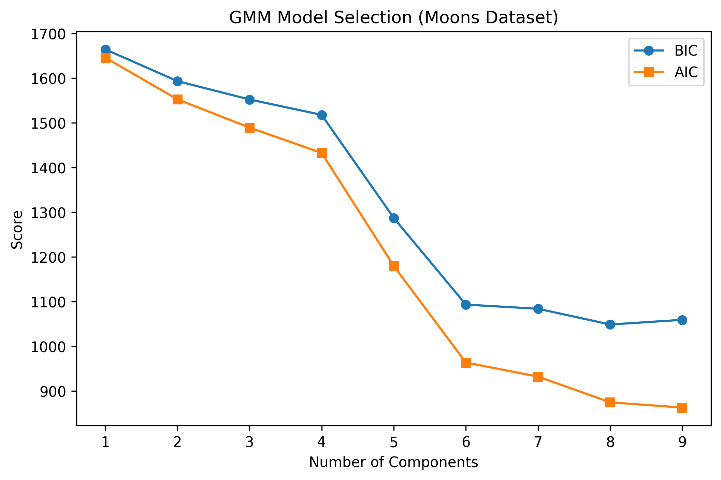
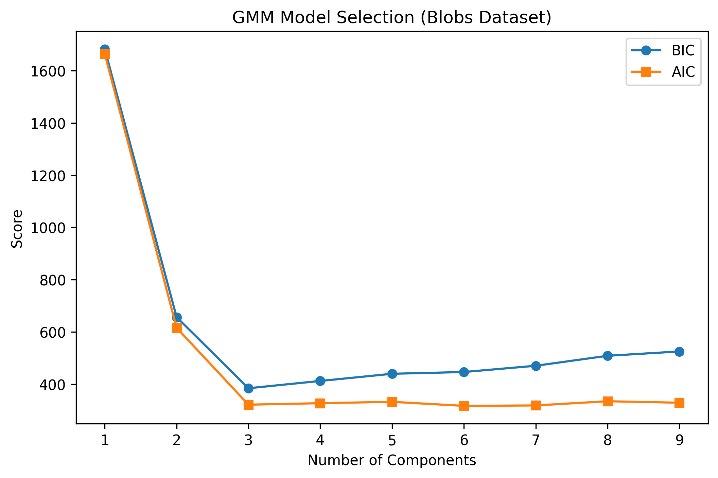
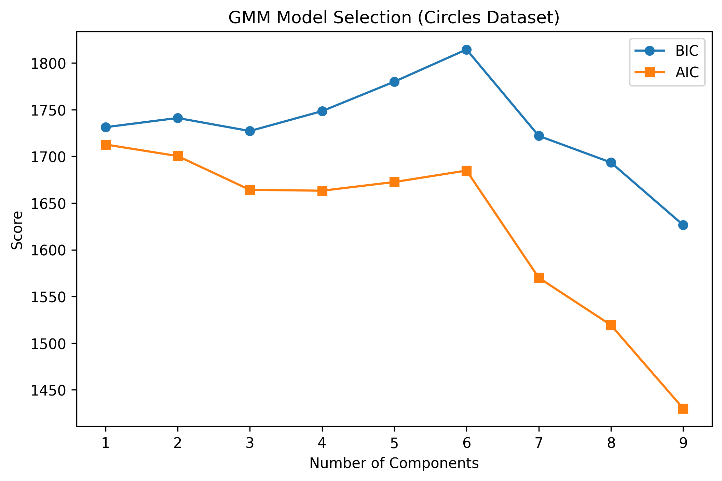
The **Blobs Dataset** shows that **k-Means and GMM produce nearly identical results**, indicating well-separated spherical clusters

**

The **Circles Dataset** highlights **GMM’s advantage over k-Means**, as it can model circularly distributed data more effectively.

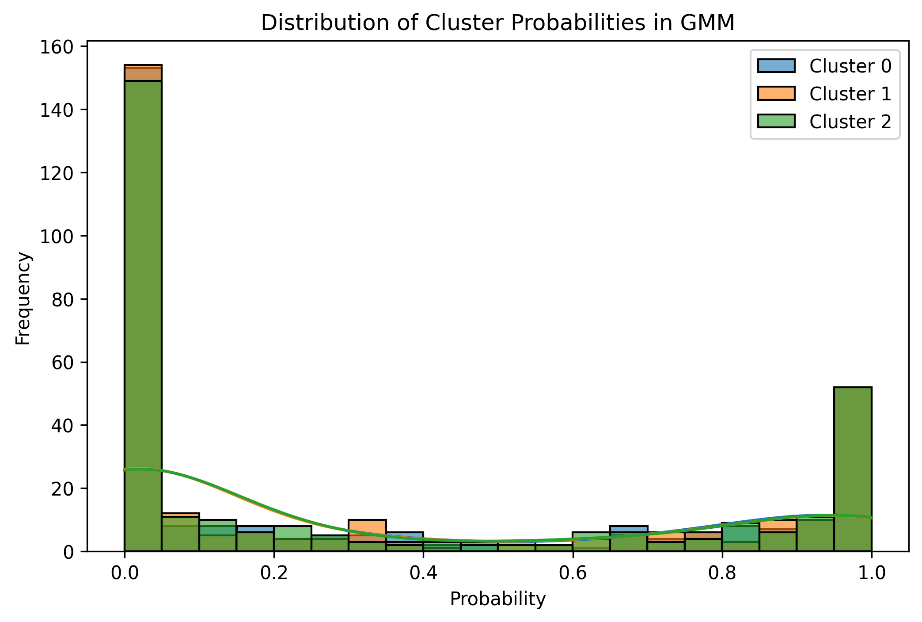
**Silhouette Score Comparisons:**k-Means scores highest on the Blobs Dataset, confirming its suitability for spherical clusters. DBSCAN struggles with the Circles Dataset, reinforcing its limitations in handling concentric patterns. GMM consistently provides comparable performance to k-Means but with added flexibility.

**Fig. 2: BIC & AIC Scores for GMM Model Selection:** *Graphs displaying Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) scores for selecting the optimal number of Gaussian components in GMM.*



* The **Blobs Dataset** correctly identifies 3 as the optimal number of clusters, confirming GMM’s ability to find the correct model complexity.
* The **Moons and Circles Datasets** require higher BIC/AIC values, indicating more complexity is needed to model the non-linear structures.

**Fig. 3: Cluster Probability Histogram:** *Histogram displaying the distribution of cluster membership probabilities assigned by GMM, showcasing the uncertainty in clustering decisions.*

****

The histogram shows that **most points have a probability close to 0 or 1**, meaning GMM assigns them confidently to a single cluster. However, **some points have intermediate probabilities**, suggesting they are near the cluster boundaries and GMM recognizes this uncertainty.

These figures provide a comprehensive view of how GMM compares against traditional clustering methods.

* **GMM performed well on Moons Dataset** by correctly capturing the overlapping structure.
* **k-Means excelled on Blobs Dataset**, indicating its effectiveness for spherical clusters.
* **DBSCAN struggled with Circles Dataset**, highlighting its limitations with concentric patterns.
* **BIC/AIC helped optimize GMM**, whereas other methods require manual tuning of cluster numbers.

## 6. Conclusion

Gaussian Mixture Models provide a powerful alternative to traditional clustering methods by allowing soft assignments and flexible cluster shapes. While computationally expensive, GMM outperforms k-Means and DBSCAN in scenarios where clusters have **complex, overlapping distributions**. The choice of clustering technique should depend on the dataset structure and computational resources available.