

**Assignment - Part A: Task: Exploring the Connection Between K-Means and GMM**

**By:** Samia Hassouna, Saher Abu Elkheir, and Rana Al-Najjar

**For:** Dr.Ibrahim Radwan

---

**Tasks Table**

Name	Task
Samia Hassouna	Write report
Saher Abu Elkheir	Coding
Raha Al-Najar	Analyse results

K-Means and Gaussian Mixture Models (GMM) are both popular clustering techniques used in unsupervised machine learning. Here’s a comparison of the two methods, along with their characteristics, strengths, and weaknesses:

**K-Means Clustering**

**Overview:**

- K-Means is a centroid-based clustering algorithm that partitions data into K distinct, non-overlapping subsets (clusters).
- Each cluster is represented by the mean of the points assigned to it.

**How It Works:**

1. Choose the number of clusters (K).
2. Randomly initialize K centroids.
3. Assign each data point to the nearest centroid.
4. Recalculate the centroids based on the assigned points.
5. Repeat the assignment and centroid calculation until convergence (i.e., centroids do not change significantly).

### Strengths:

- Simple and fast, especially for large datasets.
- Works well with spherical clusters and when clusters are of similar sizes.

### Weaknesses:

- Requires the number of clusters (K) to be specified in advance.
- Sensitive to the initial placement of centroids and may converge to local minima.
- Poor performance on clusters with varying shapes and densities.

### When to Use K-Means?

1. **Well-Separated Clusters:** Effective when clusters are distinct and spherical.
2. **Large Datasets:** Computationally efficient for handling large amounts of data.
3. **Known Number of Clusters:** Ideal when you have prior knowledge of how many clusters (K) to use.
4. **Simple Interpretability:** Results are straightforward, with each cluster represented by a centroid.
5. **Feature Engineering:** Useful for clustering data after dimensionality reduction.
6. **Image Compression:** Can cluster colors in images to reduce color variance.
7. **Market Segmentation:** Effective for dividing customers into distinct groups based on behavior.
8. **Anomaly Detection:** Identifies outliers by locating points far from centroids.
9. **Document Clustering:** Organizes documents based on feature similarities in NLP.

## Gaussian Mixture Model (GMM)

### Overview:

- GMM is a probabilistic model that assumes that the data points are generated from a mixture of several Gaussian distributions, each representing a cluster.
- It estimates the parameters (mean, covariance) of the Gaussian distributions.

### How It Works:

1. Choose the number of clusters (components).
2. Initialize parameters for each Gaussian (mean, covariance, and mixing coefficients).
3. Use the Expectation-Maximization (EM) algorithm:
  - **E-step:** Calculate the probability that each data point belongs to each Gaussian.
  - **M-step:** Update the parameters of the Gaussians based on the probabilities calculated in the E-step.
4. Repeat the E-step and M-step until convergence.

### Strengths:

- Can model elliptical clusters and handle varying cluster shapes and densities.
- Provides a probabilistic framework, giving more information about cluster membership (soft clustering).

### Weaknesses:

- More complex and computationally intensive than K-Means.
- Requires the number of clusters to be specified.
- Can be sensitive to initialization and may converge to local optima.

### When to Use GMM?

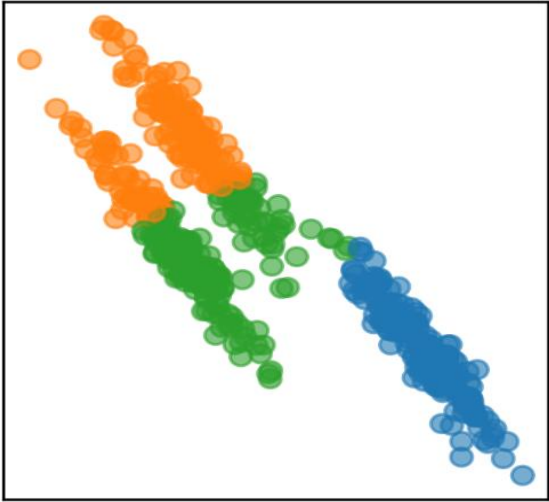
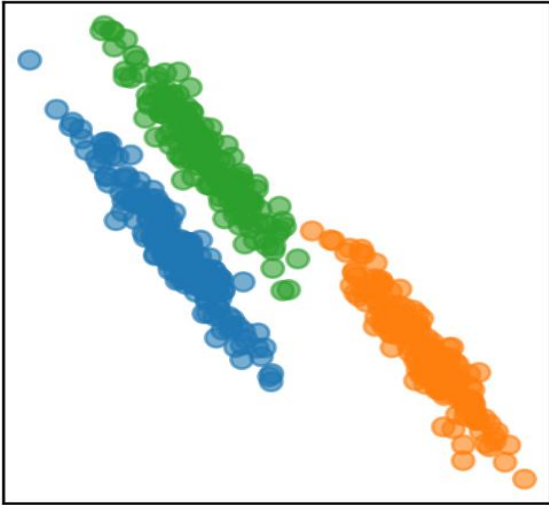
1. **Non-Spherical Clusters:** Ideal for modeling elliptical or irregularly shaped clusters.
2. **Clusters of Varying Sizes:** Can handle clusters with different sizes and densities due to its flexibility in covariance structures.
3. **Probabilistic Clustering:** Provides soft clustering, allowing data points to belong to multiple clusters with associated probabilities.

4. **Complex Data Distributions:** Effective for datasets that can be modeled as mixtures of multiple Gaussian distributions.
5. **Anomaly Detection:** Identifies outliers by detecting data points with low probability under the GMM.
6. **Feature Learning:** Useful in unsupervised learning for capturing the underlying data distribution.
7. **Image Segmentation:** Applies in image processing to segment images based on color distributions.
8. **Speech and Audio Processing:** Commonly used in speech recognition for modeling feature vectors.
9. **Market Segmentation:** Effective for segmenting customers with complex and overlapping behaviors.

## Comparison between K-Means and GMM

Feature	K-Means	Gaussian Mixture Model (GMM)
<b>Type of Clustering</b>	<b>Hard clustering:</b> <ul style="list-style-type: none"> <li>meaning that each data point is assigned to exactly one cluster.</li> <li>A point belongs to the cluster whose centroid is the nearest, and the distance metric used is typically Euclidean distance.</li> <li>No overlap between clusters; each point is strictly assigned to one cluster.</li> </ul>	<b>Soft clustering:</b> <ul style="list-style-type: none"> <li>meaning that each data point has a <b>probability</b> of belonging to each cluster.</li> <li>Instead of belonging to just one cluster, a point can partially belong to multiple clusters, with associated probabilities (or responsibilities).</li> <li>This probabilistic approach allows for <b>overlapping clusters</b>.</li> </ul>
<b>Shape of Clusters</b>	<ul style="list-style-type: none"> <li>Assumes that clusters are <b>spherical</b> and equally sized (same variance in all directions).</li> <li>Relies on the notion that clusters should be well-separated by Euclidean distance.</li> <li>Doesn't model the underlying data distribution; it simply minimizes the distance between points and centroids.</li> </ul>	<ul style="list-style-type: none"> <li>Assumes that each cluster follows a <b>Gaussian distribution</b> (or normal distribution), allowing for clusters of <b>different shapes, sizes, and orientations</b>.</li> <li>Can model clusters with <b>elliptical</b> shapes due to its ability to adjust the covariance matrix for each cluster.</li> <li>GMM explicitly models the underlying data distribution as a mixture of Gaussian distributions.</li> </ul>
<b>Initialization Sensitivity</b>	Yes	Yes
<b>Parameter Estimation</b>	Each cluster is represented by a <b>single centroid</b> (the mean of all points assigned to that cluster).	<ul style="list-style-type: none"> <li>Each cluster is represented by its <b>mean vector</b>, <b>covariance matrix</b>, and <b>mixing coefficient</b> (weight of the cluster).</li> <li>The covariance matrix allows GMM to model the shape and orientation of each cluster, providing more flexibility in the clustering process.</li> </ul>
<b>Probabilistic Output</b>	No	Yes
<b>Distance Metric</b>	<ul style="list-style-type: none"> <li>Uses <b>Euclidean distance</b> to compute the distance between data points and centroids.</li> <li>It works well when the clusters are spherical and equally sized, but struggles with elongated or non-spherical clusters.</li> </ul>	<ul style="list-style-type: none"> <li>Uses the <b>Mahalanobis distance</b> implicitly, as it takes into account the covariance of the data points when calculating the likelihood of a point belonging to a cluster.</li> <li>This makes it more adaptable to <b>irregularly shaped clusters</b>.</li> </ul>
<b>Handling Outliers</b>	is sensitive to outliers, as the algorithm tries to minimize the distance to the centroids. Outliers can distort the	GMM can handle outliers better since it considers probabilities rather than strict assignments. A point far away from a cluster

	centroid calculation, leading to suboptimal clusters.	center will have a low probability of belonging to any cluster, thus reducing its impact.
<b>Number of Parameters</b>	K-Means only needs to update the <b>centroid positions</b> , which is computationally efficient. However, it lacks flexibility as it doesn't model variance within clusters or correlations between features.	GMM is more flexible, but it also has more parameters to estimate: the <b>mean</b> , <b>covariance matrix</b> , and <b>mixing coefficient</b> for each cluster. This makes GMM more computationally intensive than K-Means, but also more powerful in modeling complex cluster shapes.
<b>Convergence Behavior</b>	Uses an iterative method based on <b>minimizing within-cluster variance</b> (sum of squared distances). It converges faster but may get stuck in local minima.	Uses the <b>Expectation-Maximization (EM) algorithm</b> , which is also iterative but more complex, as it updates the probability distributions of the clusters at each step. It converges more slowly but provides a more flexible and probabilistic interpretation of clusters.
<b>Cluster Interpretation</b>	Provides a clear and straightforward partition of the data, but the boundaries between clusters are sharp and non-overlapping.	Provides a probabilistic interpretation of clustering, where the boundaries are <b>fuzzy</b> , and a point can belong to multiple clusters with different probabilities.
<b>Speed</b>	Fast	Slower (due to EM algorithm)
<b>formula</b>	$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk}   \mathbf{x}_n - \mathbf{m}_n  ^2$ <p>J represents the sum of squares of the distance of each data point to its assigned vector mk.</p> <ol style="list-style-type: none"> <li>1. N is to total number of data points,</li> <li>2. K is the number of clusters</li> <li>3. xn is the vector of measurement n</li> <li>4. mk is the mean for cluster k</li> <li>5. rnk is an indicator variable that indicates whether to assign xn to k</li> </ol>	<p>The <b>Gaussian Mixture formula</b> for the probability density function of a data point <math>x_i</math> under GMM is:</p> $p(x_i) = \sum_{j=1}^K \pi_j \mathcal{N}(x_i   \mu_j, \Sigma_j)$ <ul style="list-style-type: none"> <li>• K is the number of Gaussian components (clusters).</li> <li>• <math>\pi_j</math> is the <b>mixing coefficient</b> for the j-th component (summing to 1).</li> <li>• <math>\mathcal{N}(x_i   \mu_j, \Sigma_j)</math> is the <b>multivariate Gaussian distribution</b> for the j-th component, defined as:</li> </ul> $\mathcal{N}(x_i   \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{d/2}  \Sigma_j ^{1/2}} \exp \left( -\frac{1}{2} (x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j) \right)$

		<p> <math>\mu_j</math>: Mean of the <math>j</math>-th component.  <math>\Sigma_j</math>: Covariance matrix of the <math>j</math>-th component.  <math>\pi_j</math>: Weight of the <math>j</math>-th component.         </p>
Example	 <p>Uses hard clustering, where each point is assigned to exactly one cluster. It assumes spherical clusters, resulting in sharp, fixed boundaries that don't capture the elongated shape of the data well.</p>	 <p>Uses soft clustering with probabilities, allowing points to belong to multiple clusters. It models elliptical clusters, adapting better to the elongated structure and overlapping areas in the data.</p>

## Implementation

In this section we will evaluate and compare the clustering performance of both algorithms using metrics like silhouette score, AIC/BIC (for GMM), and visualisation of clusters.

Before compare we need to know what is silhouette score, elbow method, and AIC/BIC.

### 1. silhouette

The silhouette score is a metric used in unsupervised machine learning to assess the quality of clustering. It measures how similar an object is to its own cluster compared to other clusters, with a score range from -1 to +1:

- **+1:** Well clustered (far from other clusters).
- **0:** On the boundary between clusters (potential misclassification).
- **-1:** Likely in the wrong cluster (closer to points in other clusters).

#### Calculation of the Silhouette Score

The silhouette score for an individual data point is calculated using the following formula:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

#### Calculation:

- The score for a point is calculated using the average distance to points in the same cluster (intra-cluster distance) and the nearest cluster (nearest-cluster distance).

#### Applications:

- Comparing clustering algorithms and determining the optimal number of clusters.
- The silhouette score can be used to evaluate the clustering quality in both **k-means** and **Gaussian Mixture Models (GMM)**

#### Limitations:



- May be misleading with varying cluster densities or shapes and less effective in high-dimensional data.

In summary, the silhouette score provides a valuable measure for evaluating cluster quality in unsupervised learning.

## 2. Elbow Method

The **Elbow Method** is a technique used in unsupervised machine learning to determine the optimal number of clusters in clustering algorithms, especially k-means.

### Steps:

1. **Select a range of  $k$  values** (number of clusters).
2. **Run the clustering algorithm** for each  $k$  and calculate the **within-cluster sum of squares (WCSS)**.
3. **Plot WCSS** against the  $k$  values.
4. **Identify the "elbow" point** where the reduction in WCSS slows down significantly. This point indicates the optimal number of clusters.

**Interpretation:** The elbow point signifies the best balance between cluster compactness and the number of clusters.

### Limitations:

- Identifying the elbow can be subjective.
- May not work well for non-convex shapes or varying densities.

## 3. AIC and BIC

**AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion)** are metrics used for model selection in unsupervised machine learning, particularly in clustering and probabilistic models like Gaussian Mixture Models (GMM).

### AIC:

- **Formula:**  $AIC = 2k - 2\ln(L)$ 
  - $K$ : Number of parameters
  - $L$ : Maximum likelihood of the model

- **Interpretation:**

- Lower AIC indicates a better model (good fit with fewer parameters).
- Used to compare models; the model with the lowest AIC is preferred.

**BIC:**

- **Formula:**  $BIC = \ln(n)k - 2 \ln(L)$

- $n$ : Number of observations
- $k$ : Number of parameters
- $L$ : Maximum likelihood of the model

- **Interpretation:**

- Lower BIC indicates a better model, with a stronger penalty for complexity than AIC, especially as sample size increases.
- The model with the lowest BIC is preferred.

## Case Study:

In our report we use **Iris** dataset and apply K-means and GMM on Iris dataset.

1. Select number of clusters: to determine suitable number of clusters we use silhouette score, AIC/BIC (for GMM).

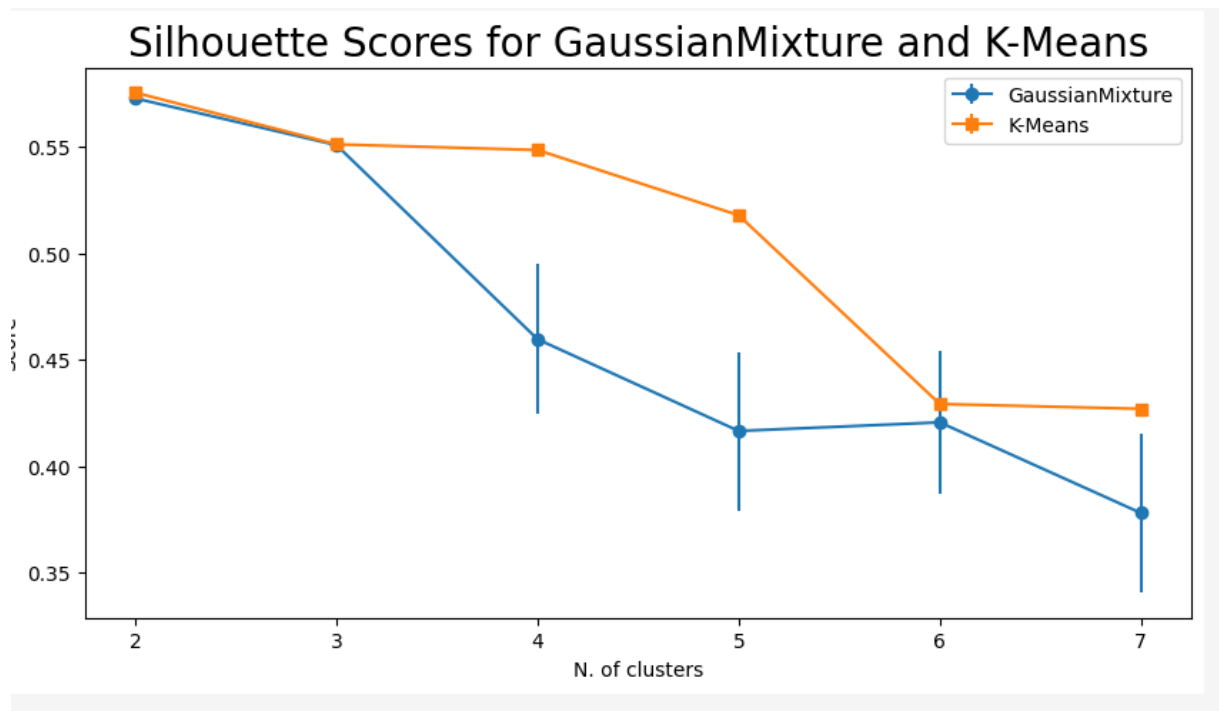


Figure 1 silhouette score for k-means and GMM

Based on the silhouette score plot for K-Means and Gaussian Mixture Models (GMM):

These selections are based on the highest silhouette scores, indicating better-defined clusters.

#### Suitable Number of Clusters:

- **For K-Means:**
  - The silhouette score is highest at **2 clusters** and decreases as the number of clusters increases. This suggests that **2 clusters** are a suitable choice for K-Means.
- **For Gaussian Mixture Models (GMM):**
  - The silhouette score steadily decreases as the number of clusters increases. It drops significantly after 3 clusters.

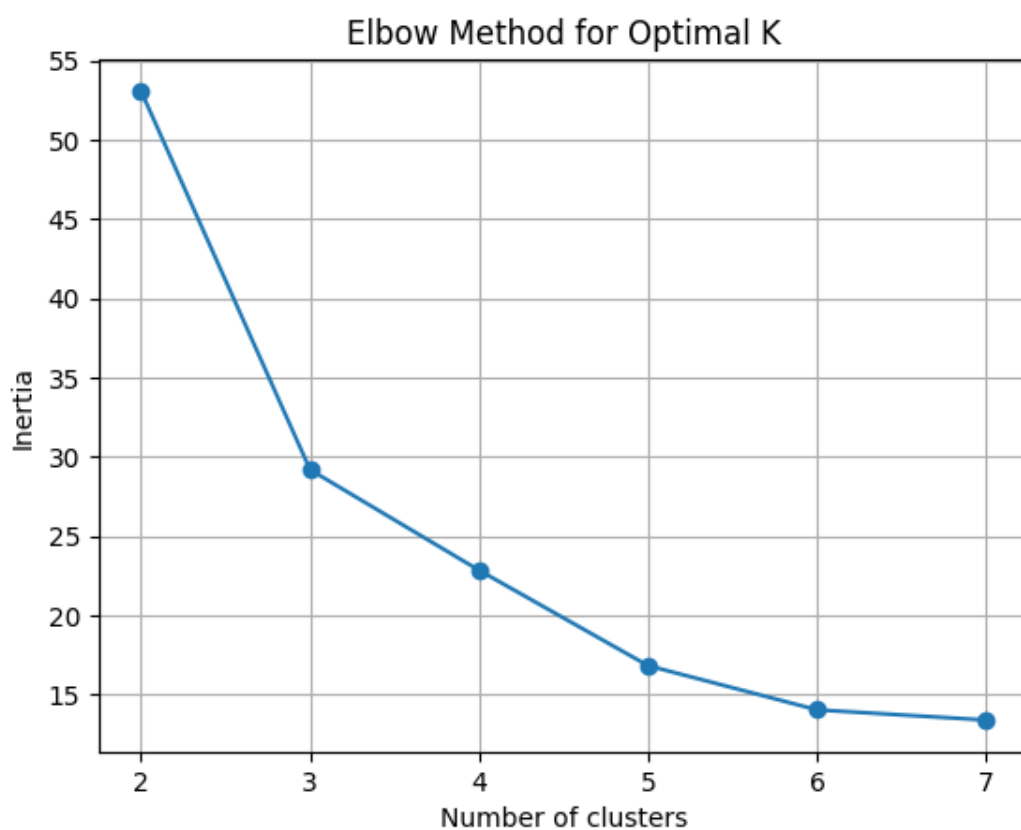


Figure 2 Elbow method for k-means

Based on the Elbow Method figure, the suitable number of clusters for K-Means is at the **elbow point**, where the rate of decrease in inertia slows down significantly. This occurs at **3 clusters**, as the curve starts to flatten after this

point. Therefore, **3 clusters** would be a good choice for K-Means based on this plot.

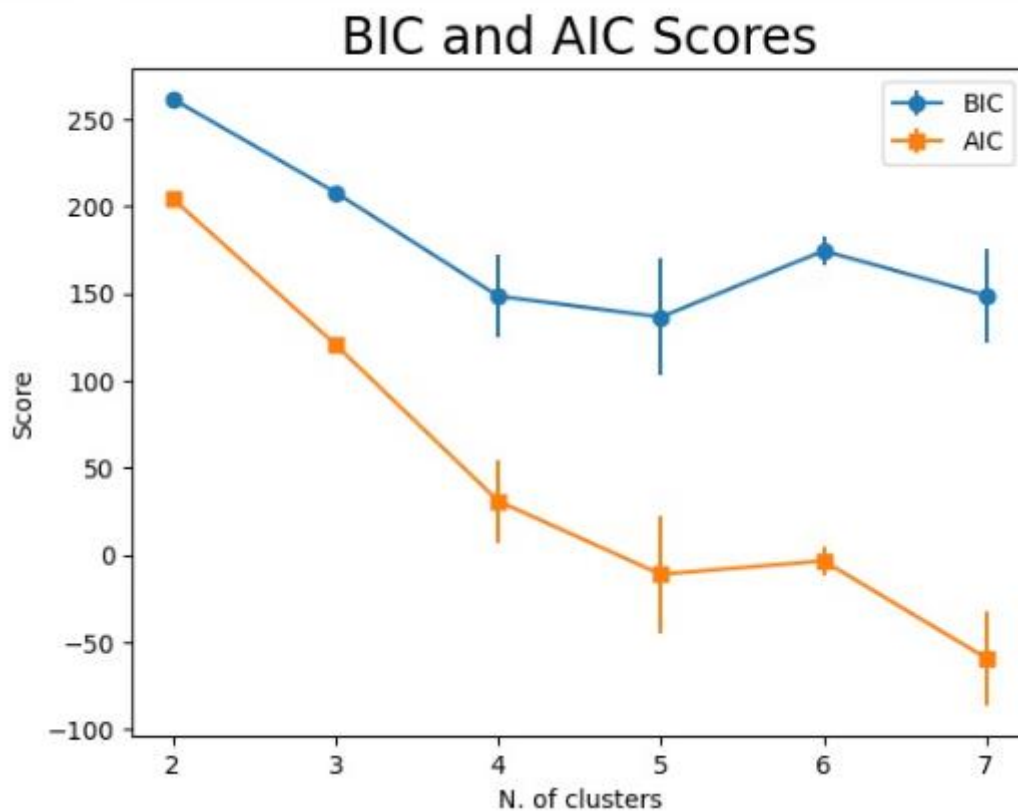


Figure 3 BIC and AIC methods for GMM

From the provided graph, which displays the Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC) scores for different numbers of clusters in a Gaussian Mixture Model (GMM), the ideal number of clusters is typically indicated by the "elbow" or point where both the BIC and AIC scores start to level off or increase after a sharp decline.

- For the AIC, the scores decrease as the number of clusters increases, but the decline slows after 4 clusters.
- For the BIC, the score is minimized at 5 clusters and increases slightly after that point.

Thus, based on the BIC score (which is often preferred for model selection in clustering), **5 clusters** appear to be the most suitable number of clusters for your GMM.

## Summary of Results:

- For **GMM**, there is a trade-off:
  - Based on **BIC**, **5 clusters** is optimal.
  - Based on **silhouette scores**, **2 clusters** is preferable.
- For **K-Means**, **2 clusters** is the most suitable option, though **3 clusters** can also be considered based on a slight performance difference.

Thus, if prioritizing silhouette scores, both **GMM** and **K-Means** would favor **2 clusters**. If focusing on BIC for GMM, **5 clusters** would be ideal.

### 2. Applying algorithms

When number of clusters = 2

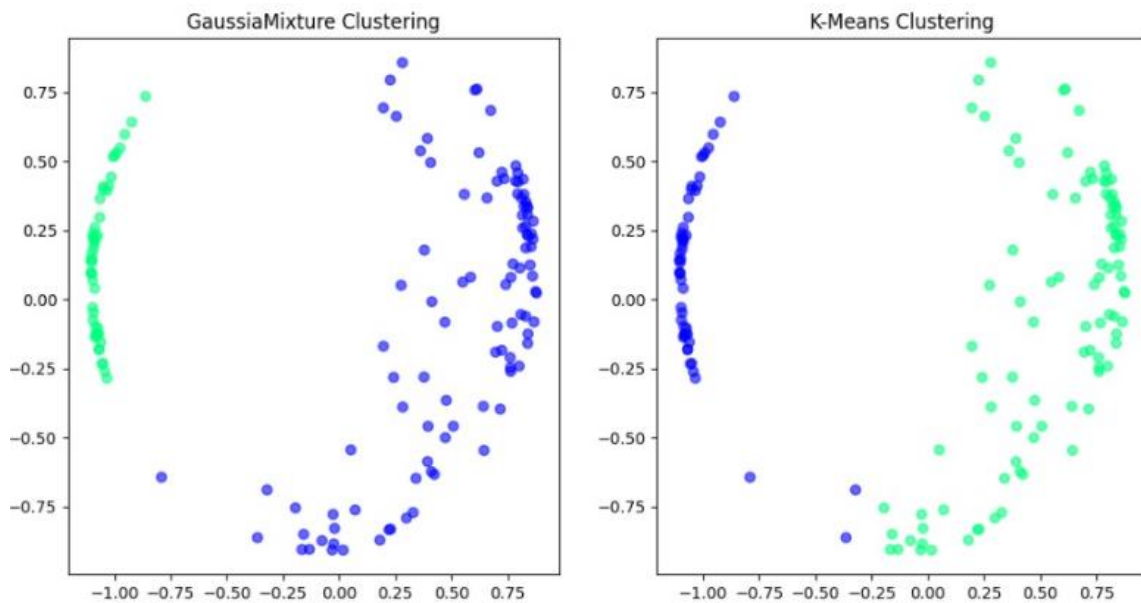


Figure 4 Apply K-means and GMM on Iris dataset with number of clusters = 2

When number of clusters = 3

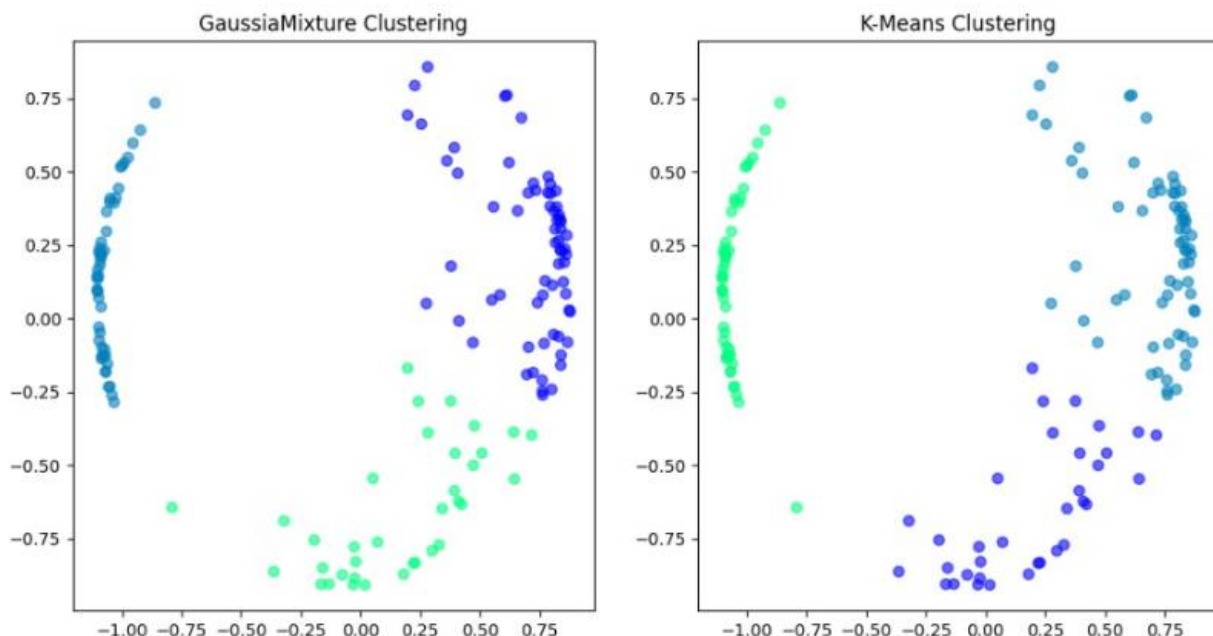


Figure 5 Apply K-means and GMM on Iris dataset with number of clusters = 3

When number of clusters = 5

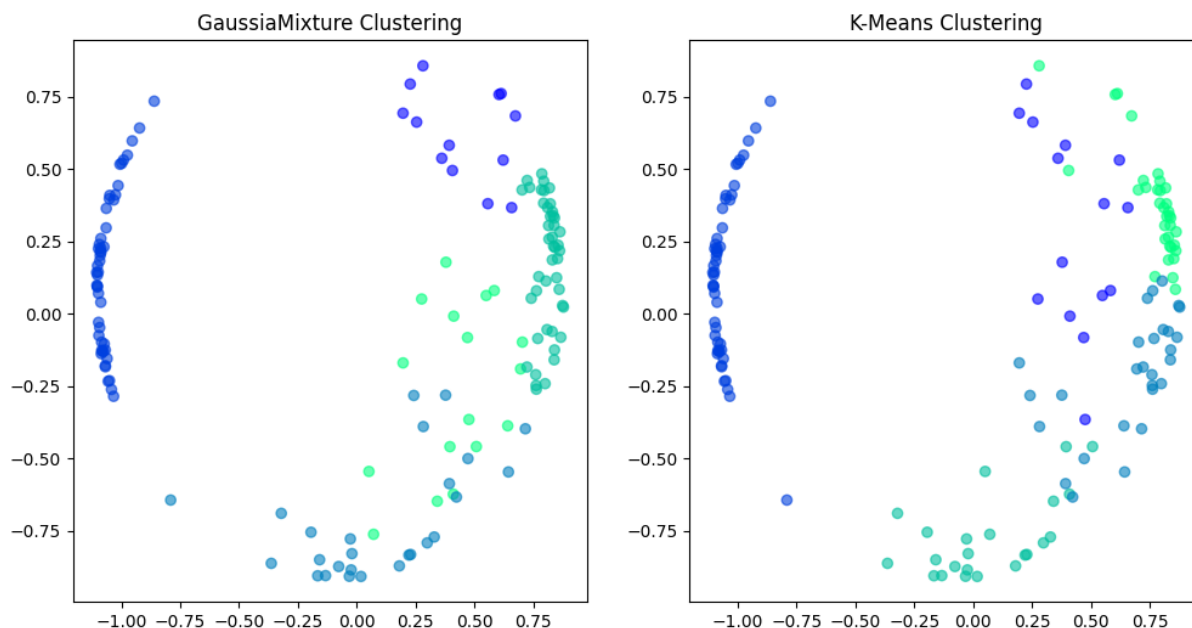


Figure 6 Apply K-means and GMM on Iris dataset with number of clusters = 5

The results compare **Gaussian Mixture Model (GMM)** and **K-Means Clustering**, both applied with 5 clusters:

- **GMM Clustering (left):**
  - Allows more flexible, elliptical cluster shapes.
  - Probabilistically assigns points, leading to smoother transitions between clusters.
  - Better captures overlapping clusters or complex shapes in data.
- **K-Means Clustering (right):**
  - Assumes rigid, spherical clusters.
  - Strictly assigns points based on distance to cluster centers.
  - Displays sharper boundaries, which might not reflect true data patterns if clusters are non-spherical.

Overall, **GMM** shows more flexibility and better handles overlapping clusters, while **K-Means** enforces simpler, clearer cluster boundaries.

## Conclusion

- **K-Means** is suitable for large datasets with well-separated spherical clusters. It is efficient and easy to implement but has limitations regarding cluster shape and size.
- **GMM** is more flexible and can capture more complex cluster structures, but it is computationally heavier and requires careful consideration of initialization and convergence.

For more details about code and implementations from this [link](#).