

K-Means vs Gaussian Mixture Models: A Practical and Visual Guide to Clustering

Author: Dekshina Udayan

Student ID: 24068646

Course: Machine Learning and

NeuralGitHubRepository: <https://github.com/dekshinaudayan/kmeans-GMM-clustering-tutorial>

1. Introduction

Clustering is a central task in unsupervised machine learning. Unlike in supervised learning, clustering does not depend on labelled examples; rather, it seeks to uncover the natural structure present in the data. Many practical problems, such as customer segmentation, anomaly detection, document grouping, and image compression all depend on clustering analysis of similarities between samples.

Two of the most common clustering methods are K-Means and GMMs. While both have the objective of clustering data points, the principles on which these algorithms are based on are immensely different.

- K-Means relies on a geometric, distance-based approach.
- GMMs use probability-based modelling, which allows them to capture the shape and structure of the data with greater flexibility.

This tutorial outlines how each algorithm works, illustrates their performance on real datasets, and discusses practical ways to select an appropriate number of clusters.

2. K-Means: Geometry-Based Clustering

K-Means is widely used because it is computationally efficient and straightforward to apply. It works best when clusters are compact in shape and clearly distinct from one another. It groups data by minimising the distance between data points and their assigned centroid.

2.1 How K-Means Works

K-Means follows an iterative procedure:

- Choose the number of clusters, K .
- Initialise K centroids at random or using `k-means++`.
- Hard assignment where every data point is assigned to the nearest centroid.
- Update the centroids by computing the mean of all points assigned to each cluster.
- Repeat steps 3–4 until centroids stop moving significantly.

This is an efficient process normally stabilising rather quickly.

2.2 Strengths of K-Means

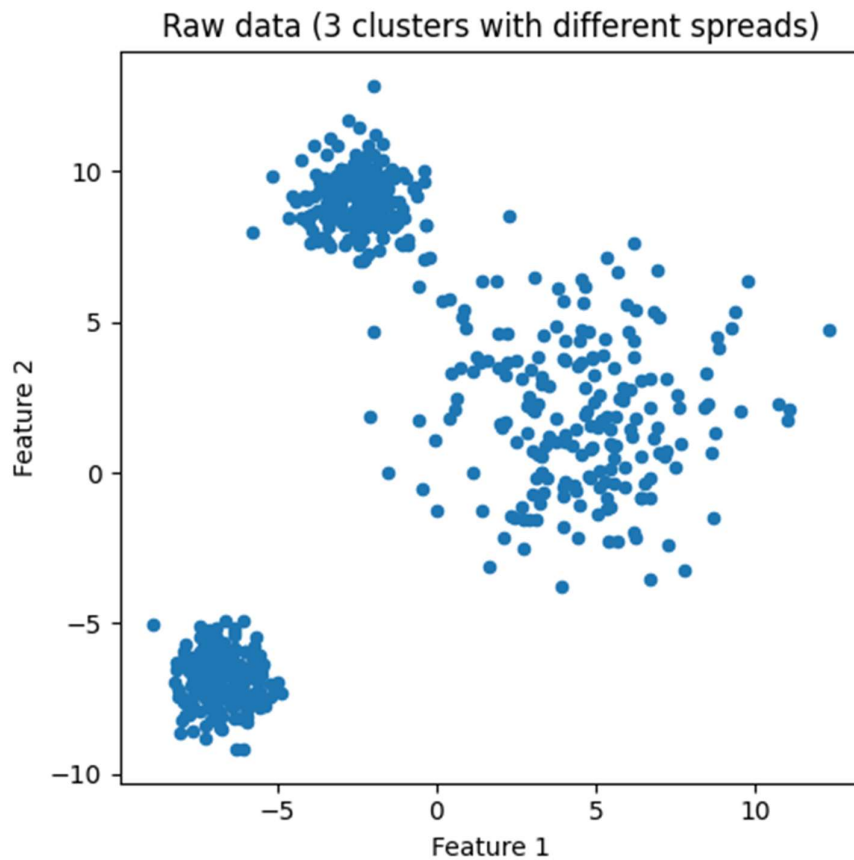
- Fast and scalable to large datasets.
- Simple and clear.
- Works well if clusters are roughly spherical and of equal size.

2.3 Limitations

- Cannot model elongated or overlapping clusters.
- Sensitive to initialisation.
- Requires knowing K in advance.
- All clusters have equal weight—cannot represent probability.

2.4 Visual Example

The figure below shows K-Means clustering on a dataset with three clusters of different spreads. The algorithm assigns each point to the nearest centroid producing crisp hard cluster boundaries.



3. Gaussian Mixture Models: Probability-Based Clustering

Gaussian Mixture Models consider a very different approach. In contrast to the method of assigning every point to some cluster, GMM takes the data as being generated from a mixture of Gaussian (normal) distributions. Each Gaussian component has:

- A mean
- A covariance matrix shape and orientation
- A mixing weight

This enables GMMs to model complex, elliptical-shaped clusters that K-Means cannot capture.

3.1 The EM Algorithm

GMMs are trained using the Expectation-Maximisation algorithm:

- E-Step: Compute the probability ("responsibility") that each point belongs to each cluster.
- M-Step: Update the means, covariances and mixing weights so that the likelihood of the data is maximized.

EM iterates between these steps until convergence.

3.2 Strengths of GMMs

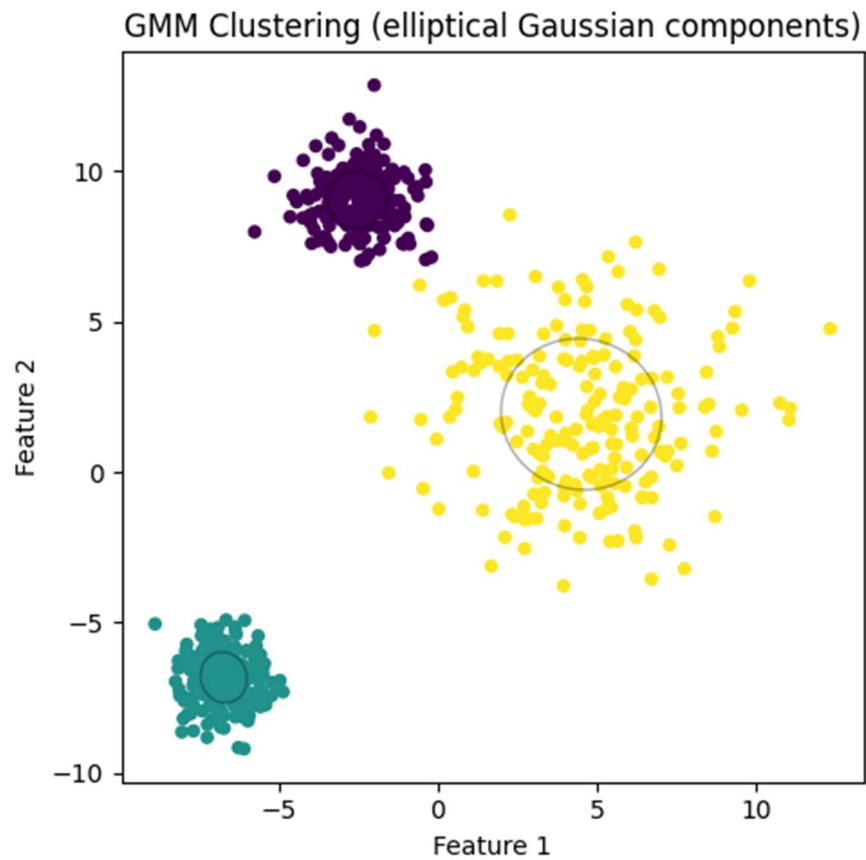
- Provides soft assignments: probabilities.
- Accounts for covariance and shape of clusters.
- More flexible and expressive compared to K-Means.

3.3 Limitations

- More computationally expensive.
- Can get stuck in local minima.
- Requires selecting the number of components.

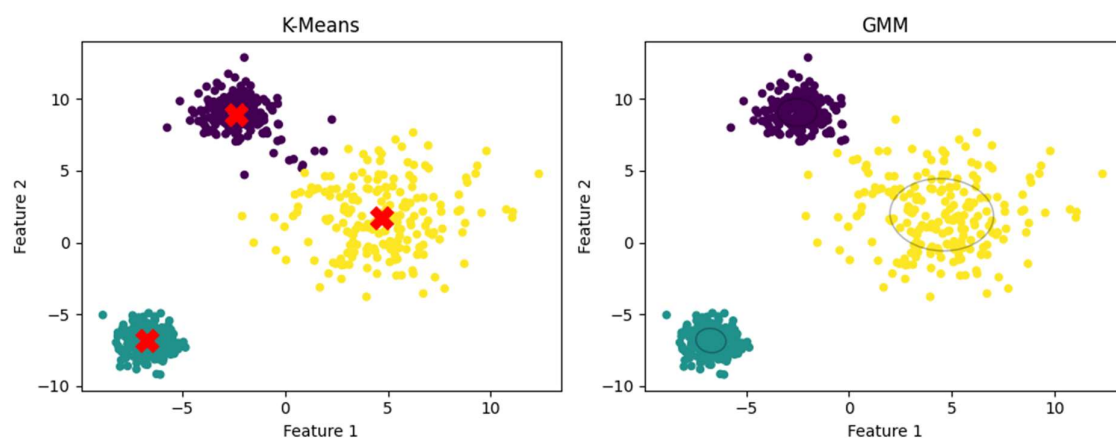
3.4 Visual Example

From the figure below it is clearly seen that the Gaussian components follow the real shape and direction of the data. Unlike K-Means which forces rigid, circular boundaries, GMM creates smooth, flexible clusters that naturally stretch and bend with the data's structure.



4. K-Means vs GMM: Side-by-Side Comparison

The following figure below demonstrates key differences:



K-Means	GMM
Hard clusters	Soft probability-based clusters
Spherical cluster assumption	Elliptical cluster assumption
Single centroid	Mean + covariance
Fast	More expensive
Poor on elongated clusters	Works very well

Interpretation of results

- In K-Means, cluster boundaries are straight, producing Voronoi-like partitions.
- In GMM, ellipses capture covariance resulting in curved and flexible boundaries.

When the cluster shape deviates from spheres, for example stretched or rotated blobs, GMM dramatically outperforms K-Means.

5.Choosing the Number of Clusters

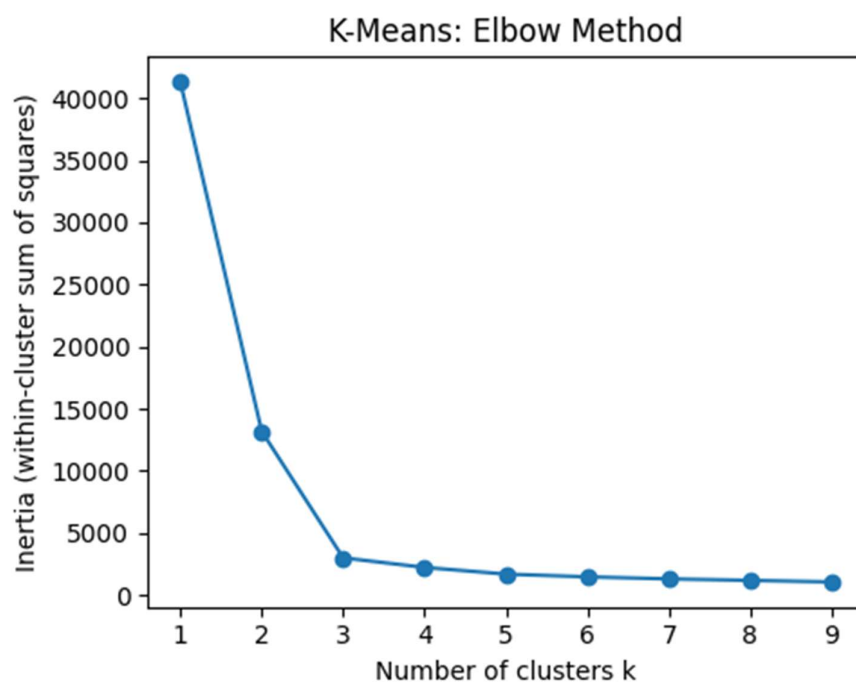
Both algorithms require choosing the number of clusters/components, K. Our method of evaluating K differs between K-Means and GMM.

5.1 Choosing K for K-Means

Elbow Method

Plot the within-cluster sum of squares (inertia) vs K.

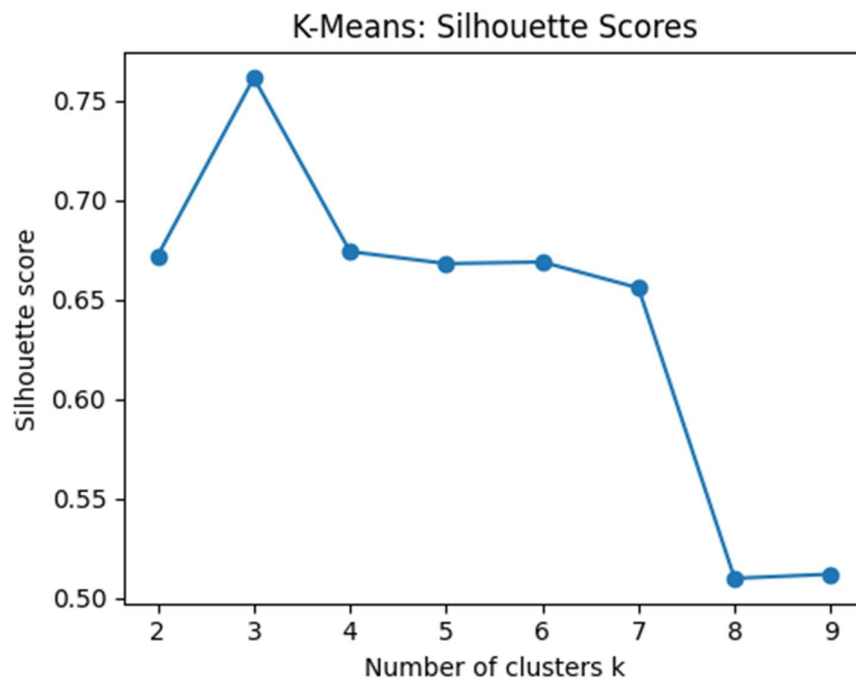
The "elbow" shows diminishing returns.



Silhouette Score

This measures how close points are to their own cluster versus others.

- Score ranges from -1 to 1 .
- Higher = better clustering.



5.2 Choosing K for GMM

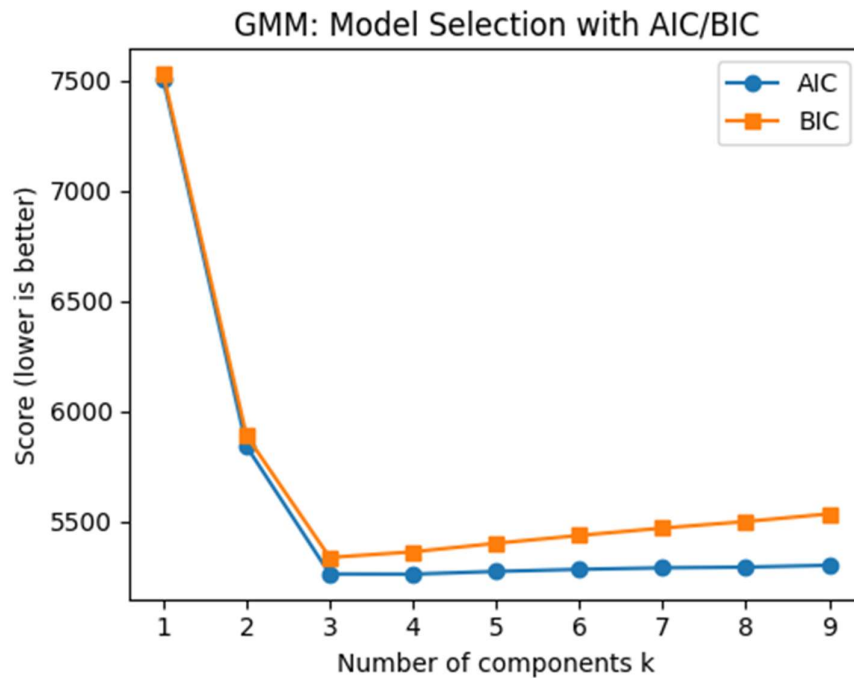
GMM provides more principled statistical criteria:

AIC: Akaike Information Criterion

BIC: Bayesian Information Criterion

Both quantify model quality while penalizing complexity.

- Lower AIC/BIC = better model.
- BIC penalizes complexity more severely compared to AIC.

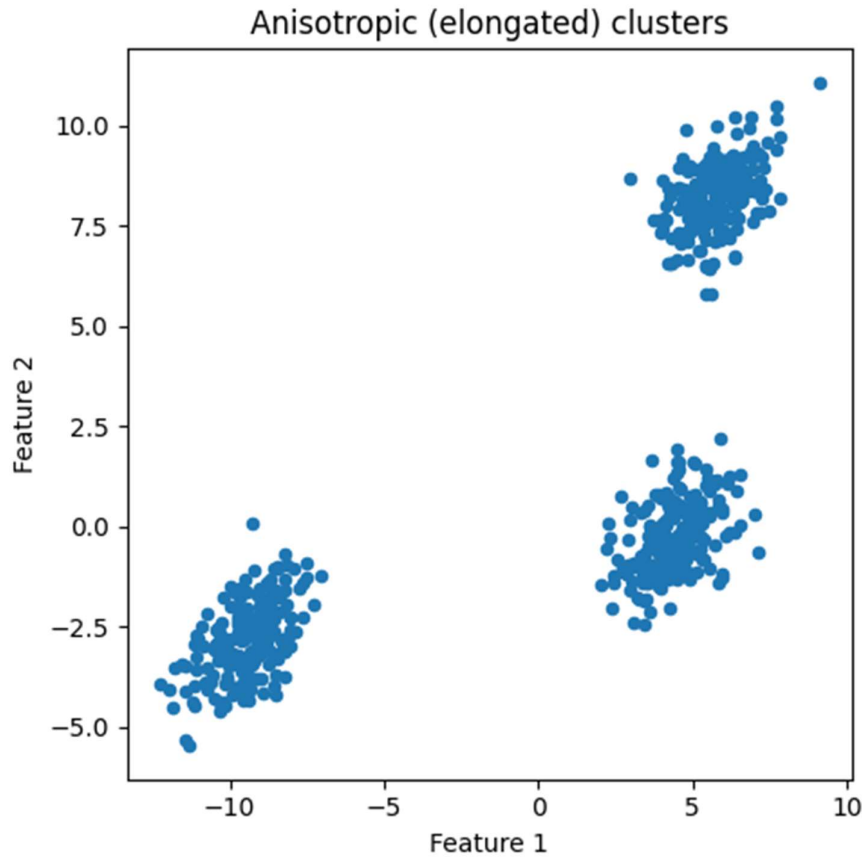


6.Failure Case: When K-Means Fails

One of the best ways to understand the difference between the two algorithms is to show when K-Means breaks down.

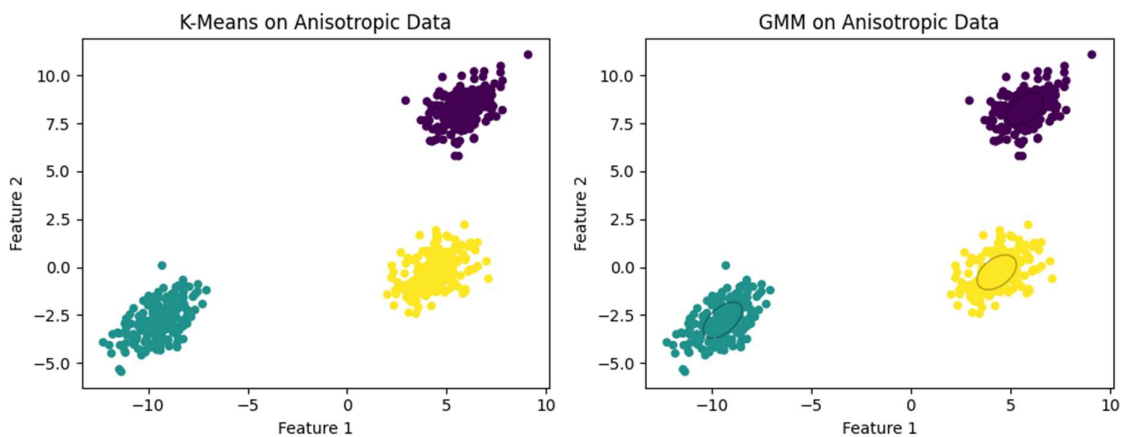
6.1 Anisotropic (Stretched) Data

When clusters are elongated or rotated, K-Means enforces circular boundaries that may not fit well to the data.



6.2 Results on Anisotropic Data

- K-Means makes many incorrect point assignments.
- GMM models capture elongated shapes via covariance matrices.



GMM is able to adjust to different cluster shapes, while K-Means is limited by its assumption of spherical groups. As a result, GMM is more suitable when the data contains clusters that are elongated or irregular.

7. When to Use Which Algorithm

Use K-Means when:

- Data is roughly spherical or well-separated.
- Speed is vital.
- You want a simple, interpretable model.
- You want hard assignments, one cluster per point.

Use GMM when:

- Clusters have different shapes and orientations.
- You need soft probability assignments.
- Data is noisy.
- You want a more flexible method for clustering.

8. Summary

This tutorial outlines both algorithms, demonstrates how they behave on real datasets, and discusses practical methods for selecting an appropriate number of clusters.

- K-Means is simple, fast but limited.
- GMMs provide more expressive modelling via covariance and probability.
- Choosing K varies in each method.
- GMMs handle non-spherical clusters better.

Together, they serve as a basis for clustering in machine learning. Understanding their behaviour equips practitioners with the ability to choose the right tool for the right problem.

9. References

- Week 3 Lecture Slides – *Machine Learning and Neural Networks*
- Dempster, A., Laird, N., & Rubin, D. (1977). *Maximum Likelihood from Incomplete Data via the EM Algorithm*.
- MacQueen, J. (1967). *Some Methods for Classification and Analysis of Multivariate Observations*.
- Scikit-learn documentation: <https://scikit-learn.org>
- Blog: “Understanding GMM and EM Algorithm”

- Blog: "K-Means Clustering Explained Clearly"