**Chapter 9: Unsupervised Learning Techniques**

**@** [**https://neptune.ai/blog/clustering-algorithms**](https://neptune.ai/blog/clustering-algorithms)

Clustering: It is the task of identifying similar instances and assigning them to clusters (groups of similar instances).

**K-Means:**

**@** [**https://www.analyticsvidhya.com/blog/2021/11/understanding-k-means-clustering-in-machine-learningwith-examples/**](https://www.analyticsvidhya.com/blog/2021/11/understanding-k-means-clustering-in-machine-learningwith-examples/)

**@** [**https://towardsdatascience.com/a-practical-guide-on-k-means-clustering-ca3bef3c853d**](https://towardsdatascience.com/a-practical-guide-on-k-means-clustering-ca3bef3c853d)

**@** [**https://towardsdatascience.com/k-means-clustering-from-a-to-z-f6242a314e9a**](https://towardsdatascience.com/k-means-clustering-from-a-to-z-f6242a314e9a)

**@** [**https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/**](https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/)

**@** [**https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a**](https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a)

**@** [**https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1**](https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1)

The K-means algorithm identifies k number of centroids and then allocates every data point to the nearest cluster while keeping the centroids as small as possible.

# The ‘means’ in the K-means refers to averaging the data; that is, finding the centroid.

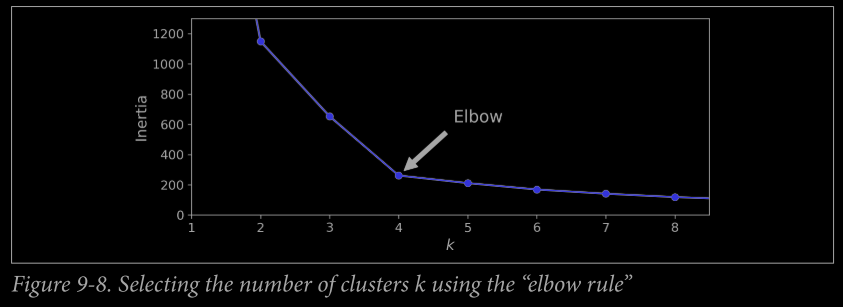
# Hard Clustering refers to assigning each instance to a single cluster, whereas Soft Clustering means assigning each instance a score per cluster.

The score can be 1) distance b/w the instance and the centroid or 2) Similarity Score (such as Gaussian Radial Basis)

# n\_init hyperparameter controls the number of random initialization centroids. By default, its value is 10.

# **Model’s Inertia** is the mean squared distance between the instance and the centroid, and the algorithm chooses the value of k which has minimum model inertia.

**But** it is not a good performance metric since it keeps getting lower as k increases.



@ <https://medium.com/analytics-vidhya/elbow-method-of-k-means-clustering-algorithm-a0c916adc540>

There is no formula for calculating the elbow point and we have to guess it by looking at graph.

# **Silhouette Score**: It is equal to mean of Silhouette Coefficients of each instance.

**Silhouette Coefficient = , -1 ≤ Silhouette Coefficient ≤ 1**

where a is the mean distance to the other instances in the same cluster

b is the mean nearest-cluster distance, that is the mean distance to the instances of the next closest cluster (defined as the one that minimizes b, excluding the instance’s own cluster).

@ <https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c#:~:text=Silhouette%20Coefficient%20or%20silhouette%20score%20is%20a%20metric%20used%20to,each%20other%20and%20clearly%20distinguished>

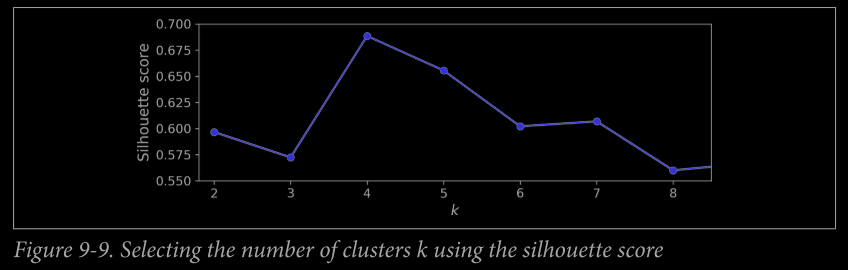
# If Silhouette Coefficient = 1 means that the instance is well inside its own cluster.

# If Silhouette Coefficient = -1 means that the instance is assigned to wrong cluster.

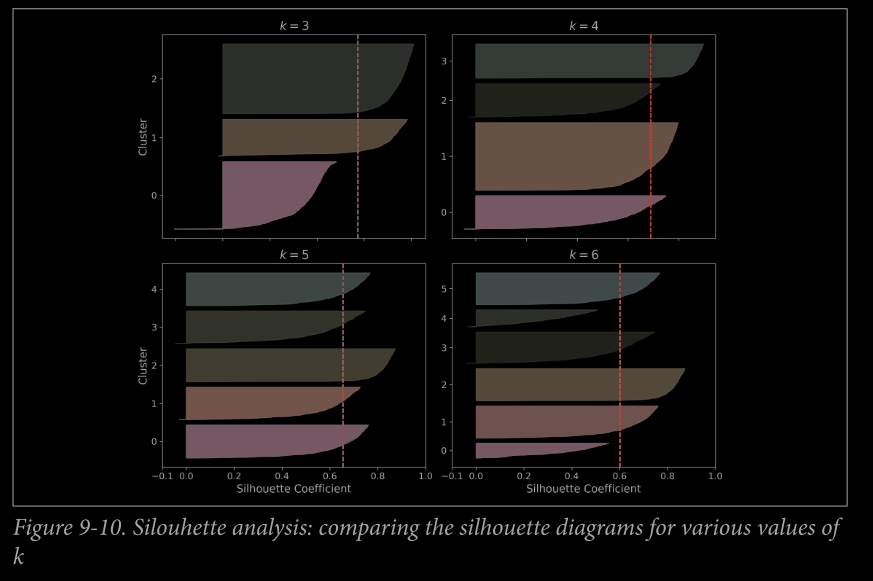
# If Silhouette Coefficient = 0 means that the instance is close to a cluster boundary.

# Best value of k is point in the graph at which Silhouette Score is maximum.

#This method is a computational expensive but precise method.



**Silhouette Diagram:** It is a graph plotted between the no. of clusters and silhouette coefficient, plotted for each training instance.



# The Vertical Red Line represents the value of Silhouette diagram.

# Model is considered as good when the peaks of all clusters cross the values of the vertical line.

E.g., from the given graph, we see model for k=4,5 peaks of clusters extends the vertical line.

@ <https://towardsdatascience.com/silhouette-or-elbow-that-is-the-question-a1dda4fb974>

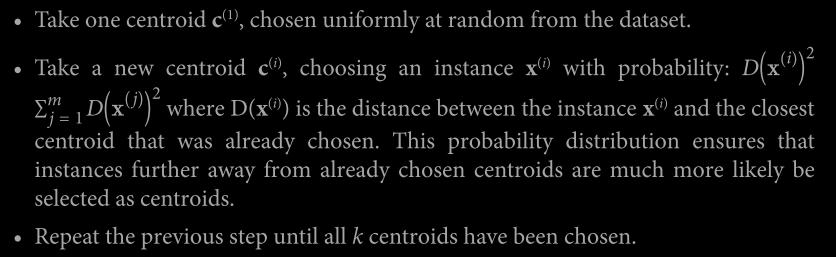
@ <https://towardsdatascience.com/elbows-and-silhouettes-hands-on-customer-segmentation-in-python-66c2e794c552>

# **Computational Complexity** of the Algorithm linearly proportion to no. of training instances(m), no. of clusters(k), and no. of dimensions(n).

# One **disadvantage of the K-means** algorithm is that it is sensitive to the initialization of the centroids or the mean points. So, if a centroid is initialized to be a “far-off” point, it might just end up with no points associated with it, and at the same time, more than one cluster might end up linked with a single centroid. Similarly, more than one centroid might be initialized into the same cluster resulting in poor clustering.

# K-Means++ initialization algorithm.

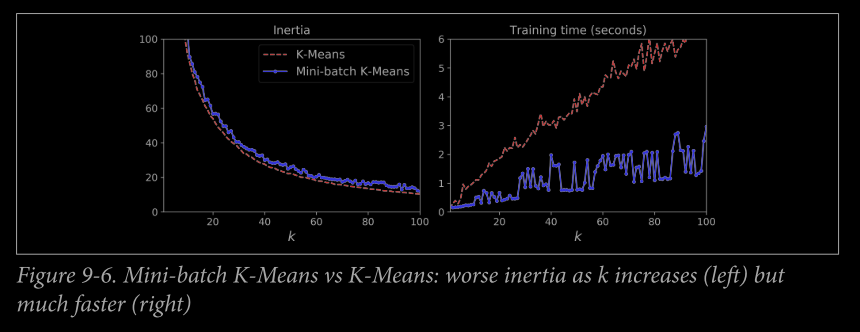
@ <https://www.geeksforgeeks.org/ml-k-means-algorithm/>



# **Mini-Batch K-Means**:

This algorithm uses mini-batches of training set(instead of the complete training set) moving the centroids just slightly at each iteration.

# Mini-batch K-Means algorithm is much faster than the regular KMeans algorithm, its inertia is generally slightly worse, especially as the number of clusters increases(k).

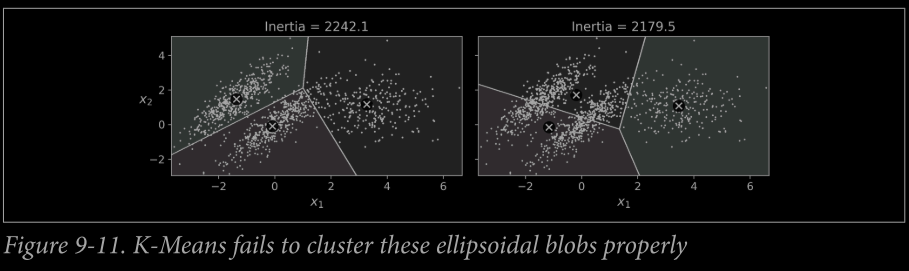


# **Limitations of K-Means**

1) We need to run the algorithm several times to avoid sub-optimal solutions.

2) We always need to specify the number of clusters.

3) K-means does behaves well when the cluster have quite varying sizes, different densities or non-spherical shapes.



**# Scaling before running the K-Means is import or else the clusters will be very stretched, and it will perform poorly.**

**Scaling doesn’t guarantees the spherical shape but is performance of model will improve.**

**## Using Clustering for Image Segmentation : READ FROM THE BOOK !!**

**Active Learning:** To continue improving the model and your training set, the next step could be to do a few rounds of active learning. This is when a human expert interacts with the learning algorithm, providing labels when the algorithm needs them.

**Uncertainty Sampling:**

**1)** The model is trained on the labelled instances gathered so far, and this model is used to make predictions on all the unlabeled instances.

**2)** The instances for which the model is most uncertain (i.e., when its estimated probability is lowest) must be labelled by the expert.

**3)** Then you just iterate this process again and again, until the performance improvement stops being worth the labelling effort.

**DBSCAN**

**@** [**https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/**](https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/)

**@** [**https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556**](https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556)

**@** [**https://towardsdatascience.com/understanding-dbscan-and-implementation-with-python-5de75a786f9f**](https://towardsdatascience.com/understanding-dbscan-and-implementation-with-python-5de75a786f9f)

This algorithm defines clusters as continuous regions of high density. It is capable of identifying any number of clusters, of any shape, it is robust to outliers, and it has just two hyperparamter (eps and min\_samples).

1) For each instance, the algorithm counts how many instances are located within a small distance ε (epsilon) from it. This region is called the instance’s ε-neighborhood.

2) If an instance has at least min\_samples instances in its ε-neighborhood (including itself), then it is considered a core instance. In other words, core instances are those that are located in dense regions.

3) All instances in the neighborhood of a core instance belong to the same cluster. This may include other core instances, therefore a long sequence of neighboring core instances forms a single cluster.

4) Any instance that is not a core instance and does not have one in its neighborhood is considered an anomaly.

# This algorithm works well if the clusters are dense enough, and they are well separated by low-density regions.

# DBSCAN class labels\_ the instance after calling fit method and if any instance’s label is -1, it means that instance is considered as **anomaly.**

# DBSCAN class does not have a predict() method, but it has a fit\_predict() method.

# KNeighborsClassifier can predict which cluster they most likely belong to, and even estimate a probability for each cluster.

# In KNN’s training set, there is no anomaly and the classifier always chooses a cluster, even when that cluster is far away.

# Computational Complexity : Aprrox O(m log(m) ) and Scikit-Learn’s implementation can require upto O( m2 ) memory is eps is lagre.

**## Other Clustering Algorithms (Pg. 259) : Agglomerative Clustering, Birch, Mean-Shift, Affinity Propagation, Spectral Clustering.**

**Gaussian Mixtures:** A Gaussian mixture model (GMM) is a probabilistic model that assumes that the instances were generated from the mixture of several gaussian distribution whose parameters are unknown.

@ <https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/>

@ <https://medium.com/swlh/gaussian-mixture-models-gmm-1327a2a62a>

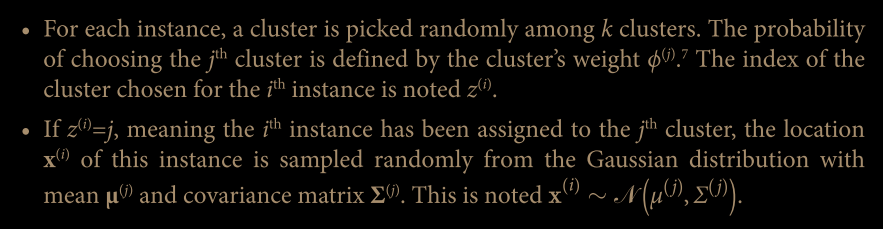
@ <https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95>

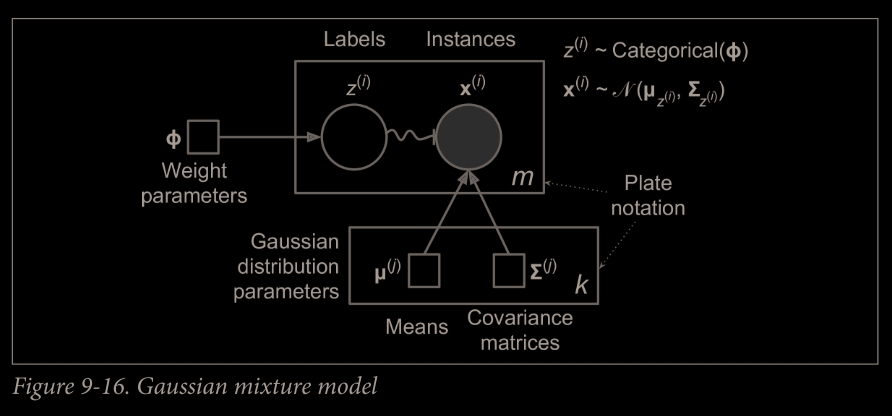
@ <https://towardsdatascience.com/gaussian-mixture-modelling-gmm-833c88587c7f>

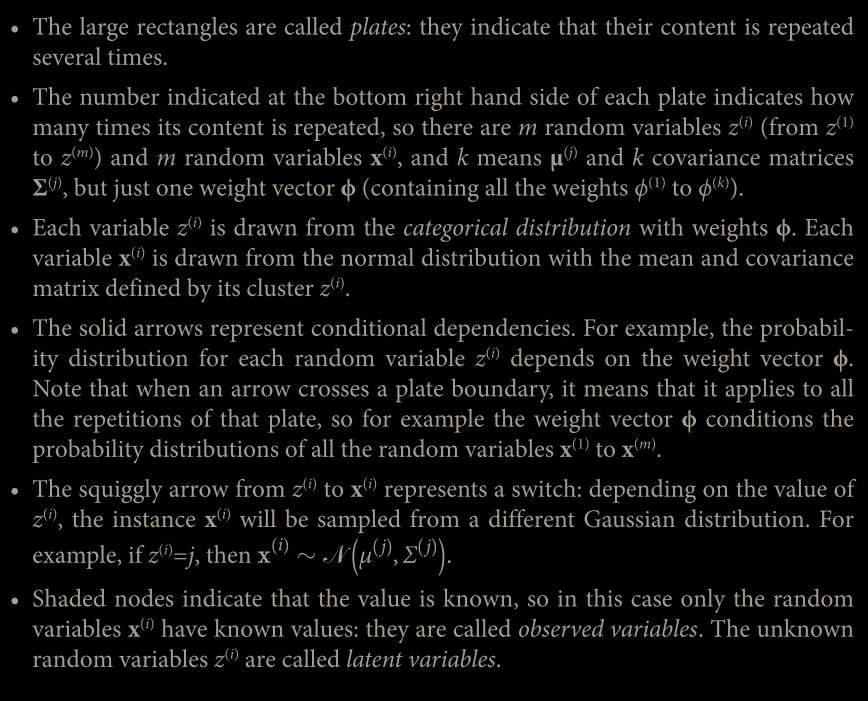
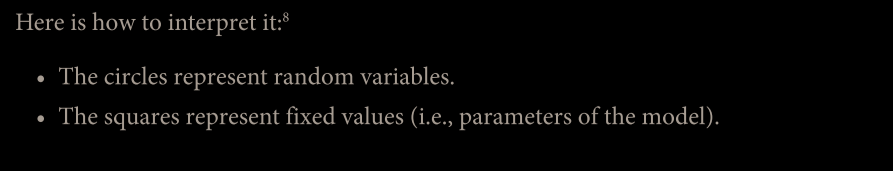
All the instance generated from a single Gaussian Distribution form a cluster that typically looks like as ellipsoid. Each cluster can have a different ellipsoidal shape, size, density and orientation.

**GaussianMixture Class:**

In this variant, no. of cluster k must be known in advance and dataset X is generated through is probabilistic process:







**Expectation-Maximization (EM) Algorithm:**

This algorithm is very similar or say generalization of K-Means algorithm, it also initializes the cluster parameters randomly, then it repeats two steps until convergence, first assigning instances to clusters (this is called the expectation step) then updating the clusters (this is called the maximization step).

# EM not only finds the cluster centers (μ’s) but also their size, shape and orientations (Σ’s) as well as relative weights (ϕ’s).

# Unlike K-Means, EM using soft clustering instead of hard clustering.

# During expectation step, EM calculates the probability of finding of each instance in a cluster.

# During maximization step, each cluster is updated using all the instances in the dataset, with each instance weighted by the estimated probability that it belongs to that cluster.

These probabilities are called the responsibilities of the clusters for the instances.

# Limitation of EM: Like K-Means, EM can also end up converging to poor solutions, so it needs to be run several times. (hypermeter n\_init is used to do so).

# EM is a generative model, which means one can sample new instances from it.

# Computational Complexity: EM cannot will be scale to large no. of features.

If covariance\_type is ‘spherical’ or ‘diag’ then O(kmn).

If covariance\_type is ‘tied’ or ‘full’ then O(kmn2 + kn3).

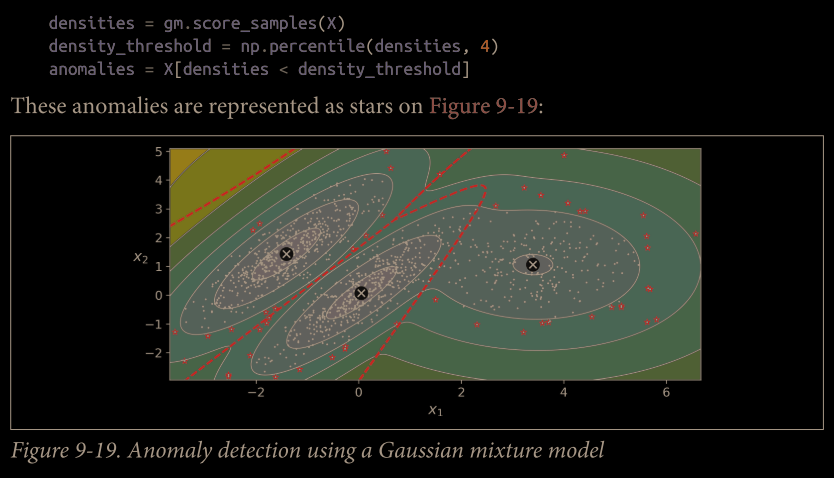
**## GO THROUGH THE BOOK FOR MORE DETAILS OF THE HYPARAMETERS, AS EVERYTHING IS NOT WRITEN HERE.**

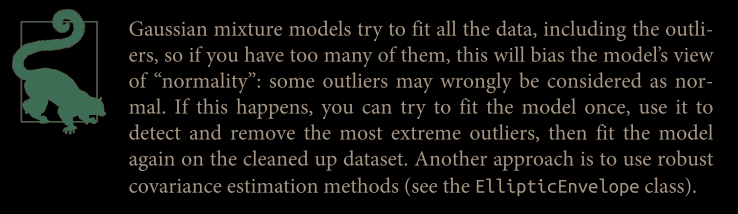
**Anomaly Detection using Gaussian Mixtures**

Anomaly Detection is a task of detecting the instances that deviate strongly from the norm.

# With the Gaussian Mixture Model, anomalies are those instances who lies in a low-density region. We have to define a density threshold what we wean to use.

# Here is how you would identify the outliers using the 4th percentile low density as the threshold (i.e., approximately 4% of the instances will be flagged as anomalies):





# We **cannot use inertia or silhouette score** metrics to select appropriate no. of clusters **for Gaussian Mixture Model** because they are not reliable when the clusters are not spherical or have different sizes.

# For gaussian mixture model, one can try to minimize a theoretical information criterion such as the Bayesian Information (BIC) or the Akaike Information Criterion (AIC).

