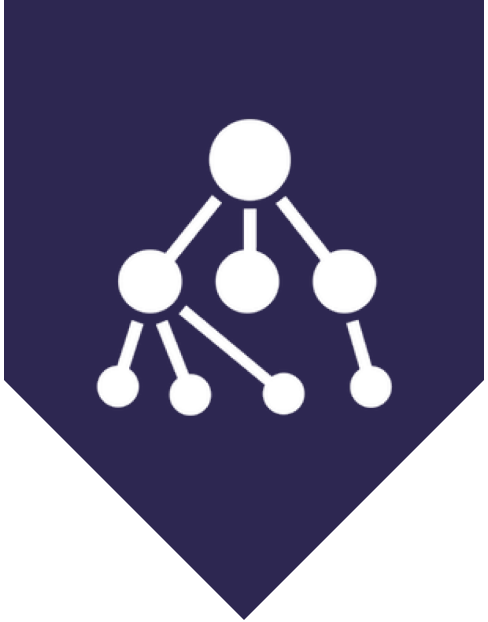


CLUSTERING ALGORITHMS

Eshan Jain
Senior Data Scientist – WalmartLabs

Organized By

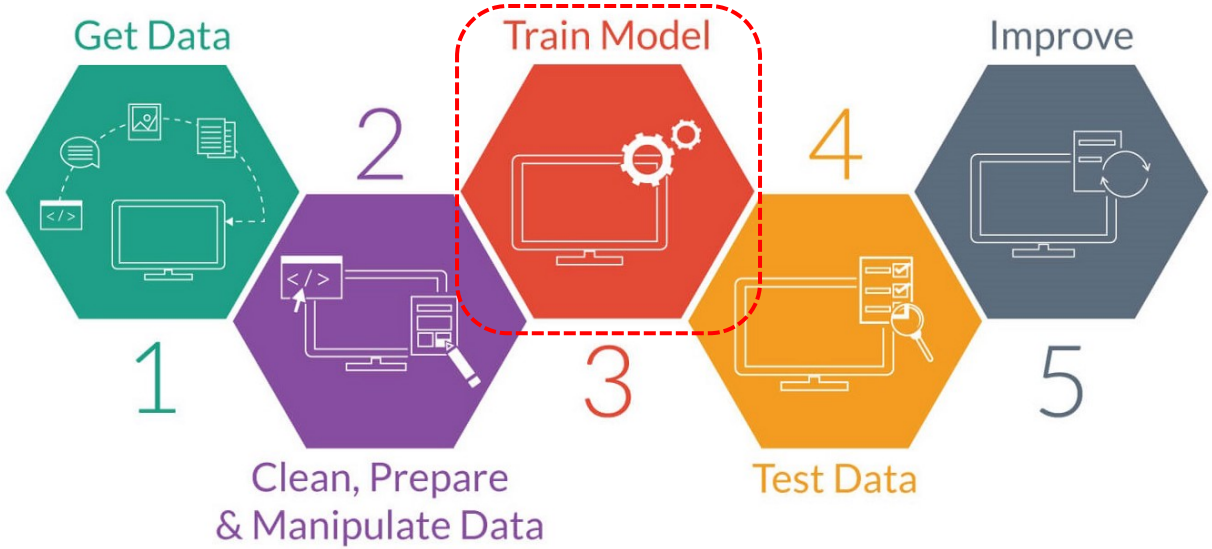
GREYATOM

Agenda

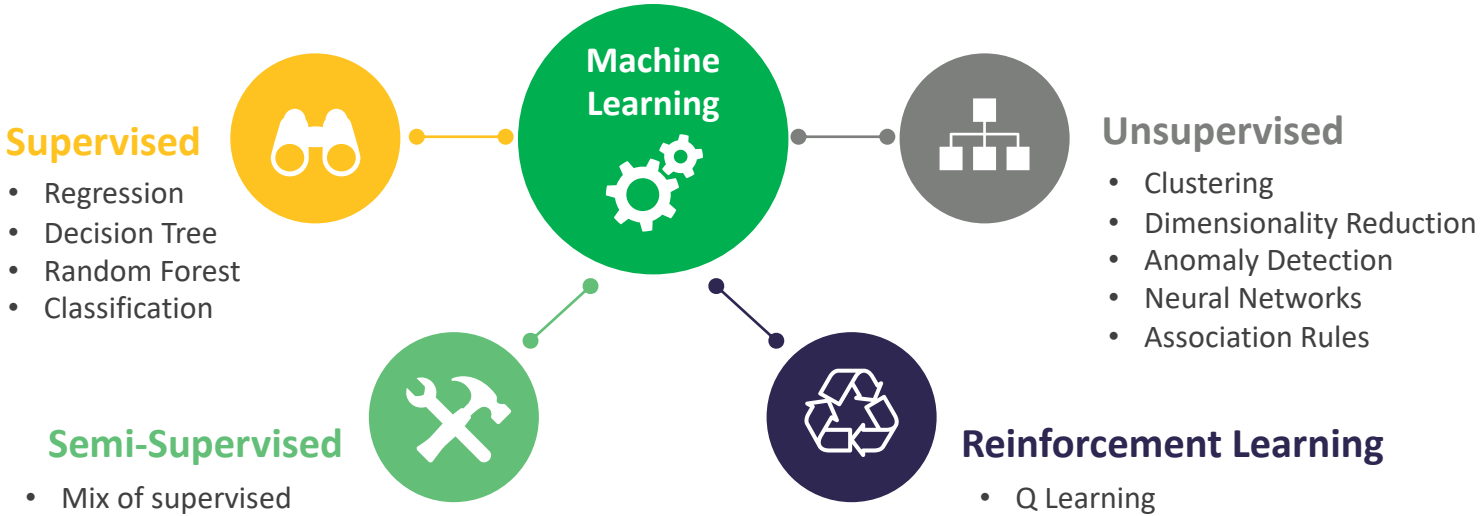
- Understand your ML problem framework
- Unsupervised Machine Learning
- Intuition behind clustering analysis
- Clustering types and working
- Industry use cases
- Code walkthrough
- Closing remarks
- Questions

What’s your problem?

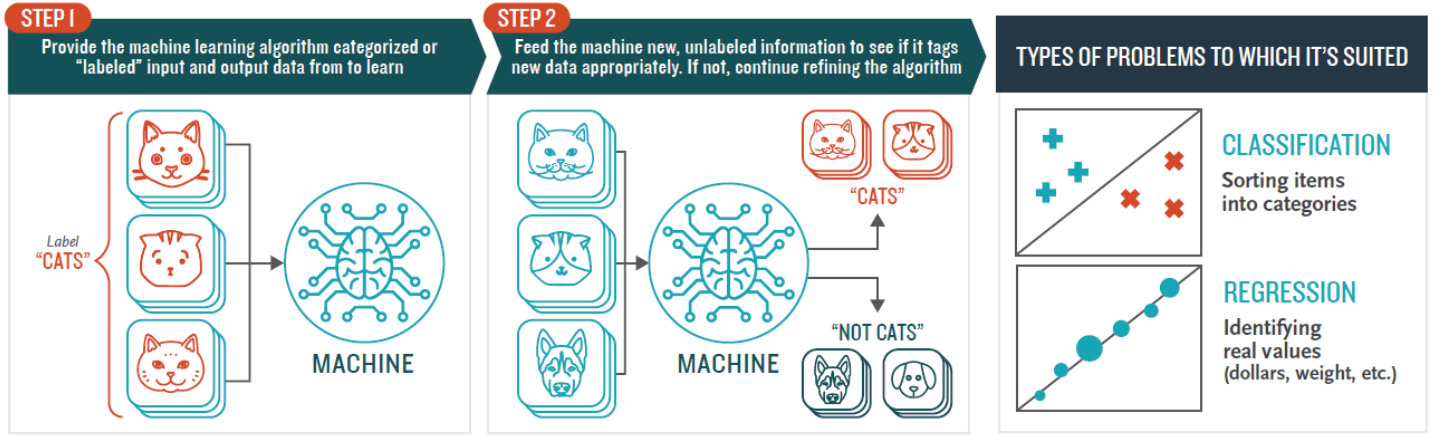
Data Scientists follow given machine learning workflow while solving most business problems on their jobs. But how do they come up with the model to be built? Let’s find out!



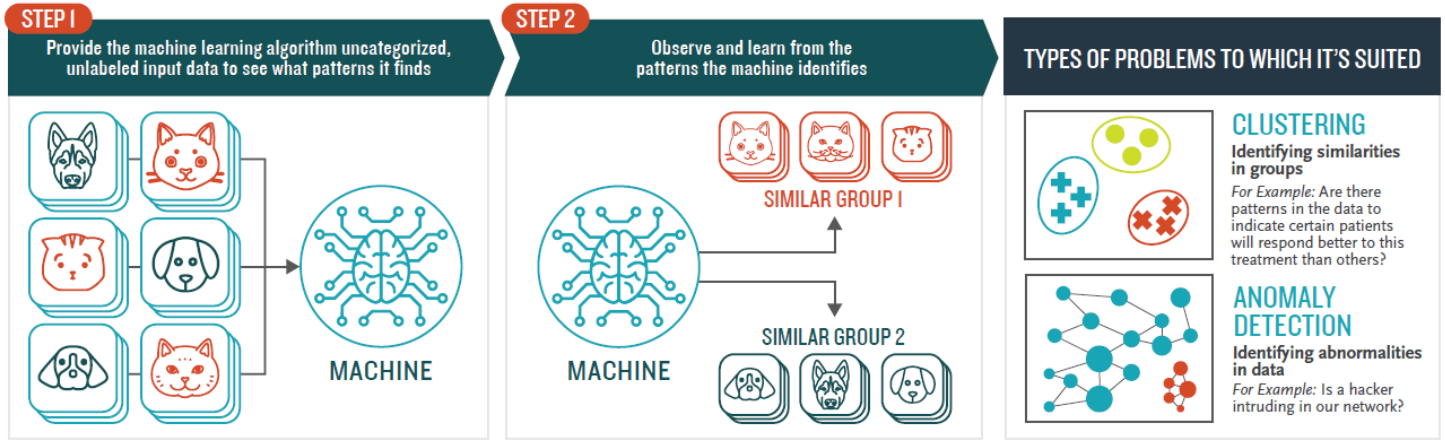
Machine Learning Algorithms



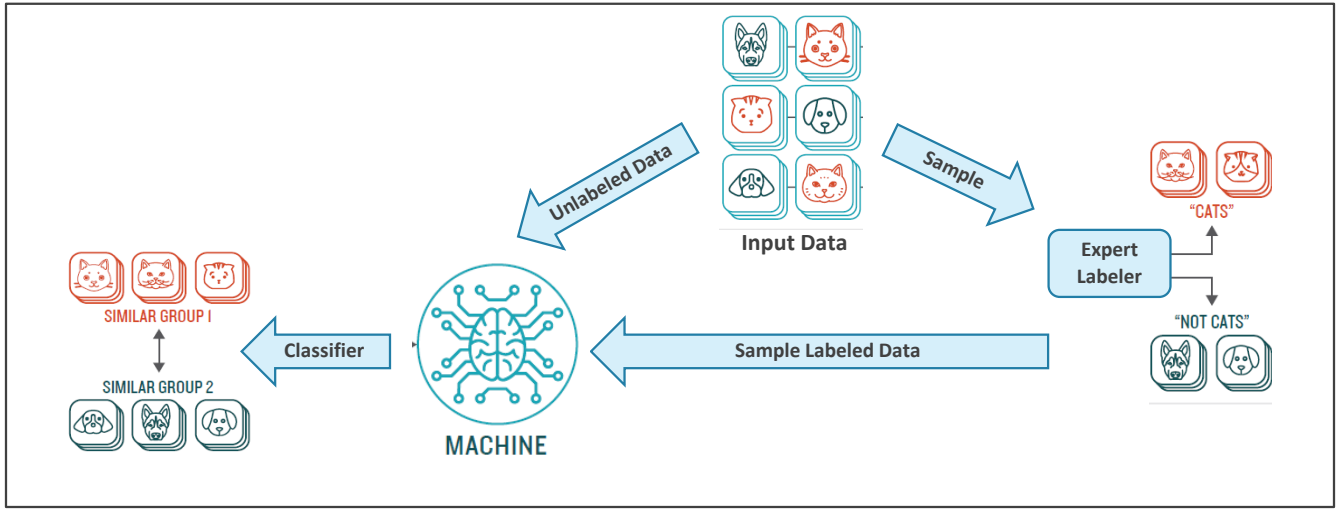
Supervised



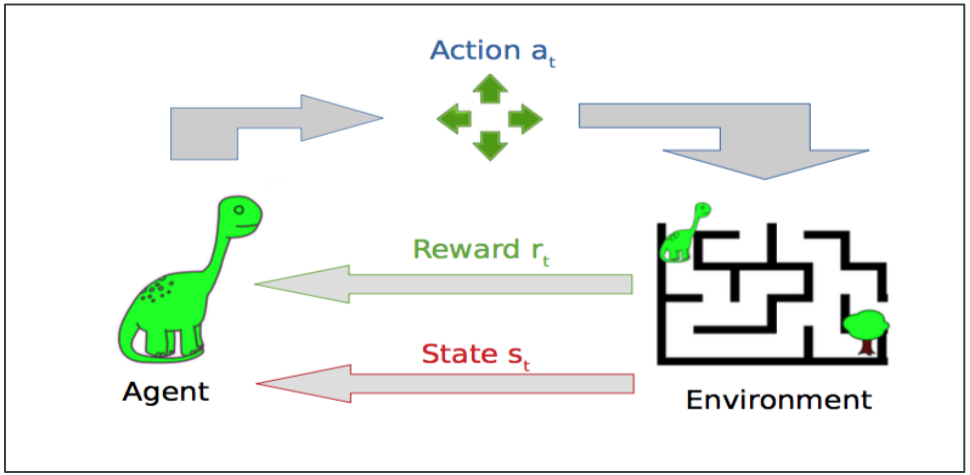
Unsupervised



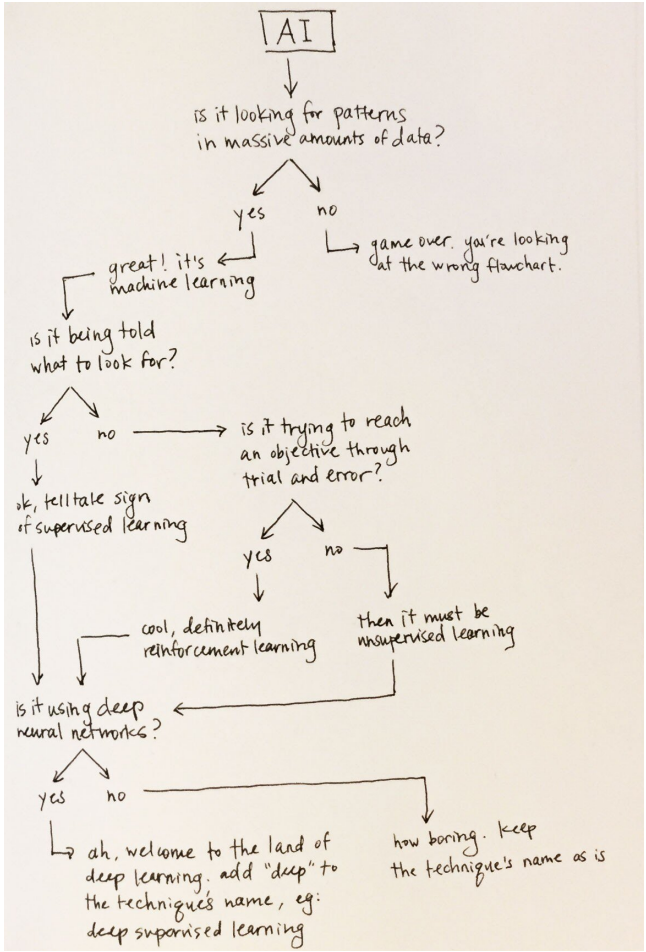
Semi-Supervised



Reinforcement Learning



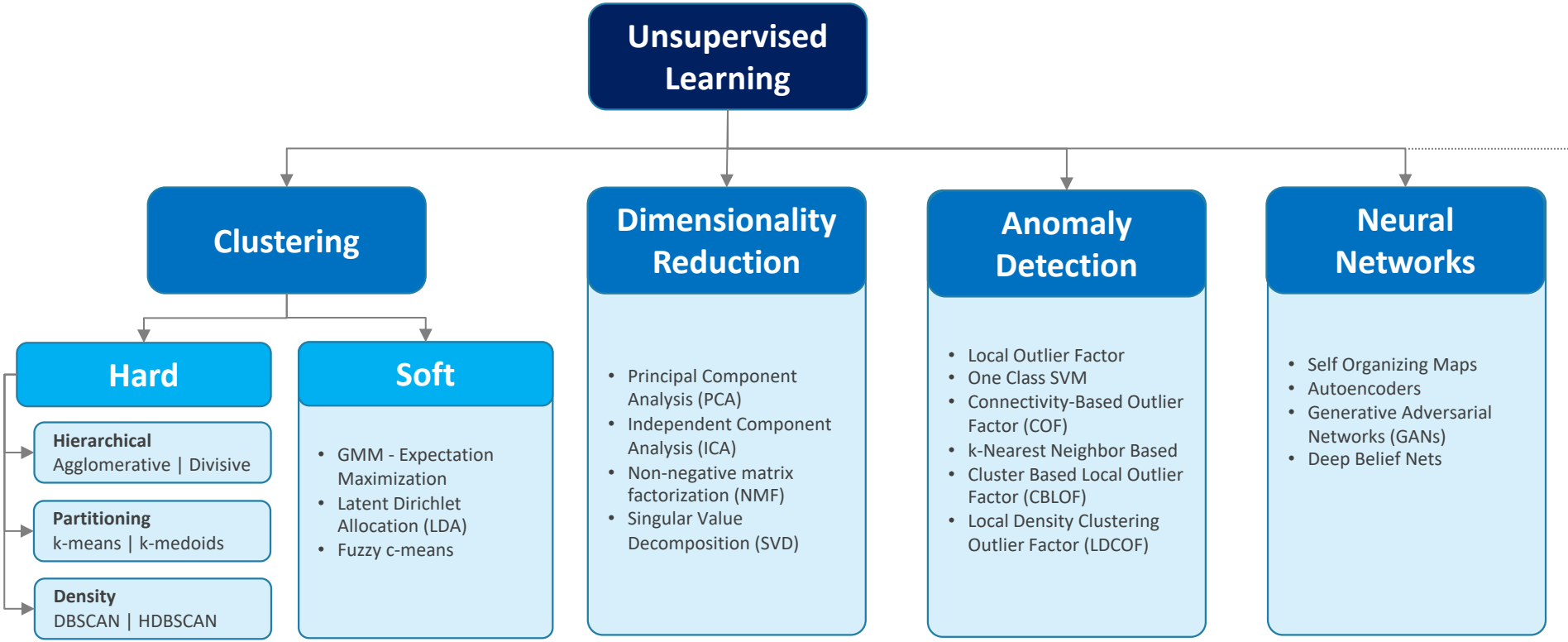
Find your problem framework!

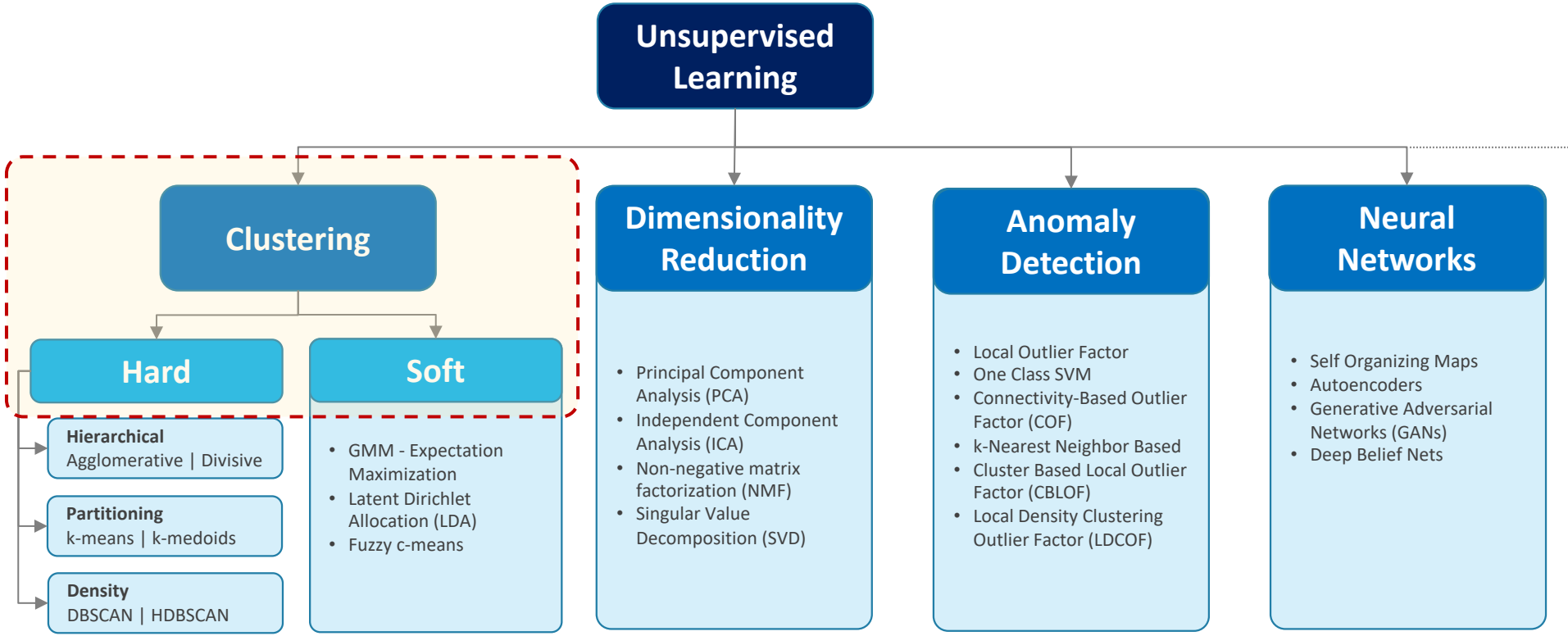


There is much more to Unsupervised Learning than k-means clustering ☺

-Unknown

Unsupervised Machine Learning




Unsupervised Learning
Clustering
Hard
Soft
Hierarchical
Agglomerative | Divisive
Partitioning
k-means | k-medoids
Density
DBSCAN | HDBSCAN
GMM - Expectation Maximization
Latent Dirichlet Allocation (LDA)
Fuzzy c-means
Dimensionality Reduction
Principal Component Analysis (PCA)
Independent Component Analysis (ICA)
Non-negative matrix factorization (NMF)
Singular Value Decomposition (SVD)
Anomaly Detection
Local Outlier Factor
One Class SVM
Connectivity-Based Outlier Factor (COF)
k-Nearest Neighbor Based
Cluster Based Local Outlier Factor (CBLOF)
Local Density Clustering Outlier Factor (LDCOF)
Neural Networks
Self Organizing Maps
Autoencoders
Generative Adversarial Networks (GANs)
Deep Belief Nets

Clustering Analysis

Clustering Analysis or Clustering *is the task of dividing the population or the data points into a number of groups such that the data points in the same group (called a cluster) are more similar to each other than to data points in other groups (clusters).*

Have you come across a business problem like-

- *Help understand customers' purchasing behavior for better targeting strategy*
- *Tell us what our customers are talking about our brand from review/social-media data*

Common Industry Usage

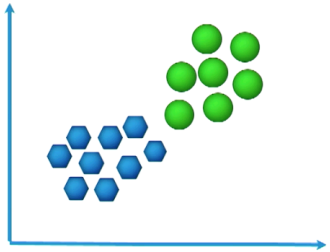
- Customer Segmentation based on demographics, transaction behavior or behavioral attributes
- Document Classification under various topics
- Automated IT Alerts

But why do we need clustering?

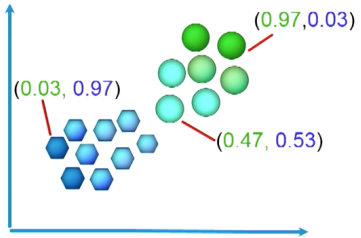
- Analytics industry is dominated by objective modelling like decision tree and regression
- Clustering generates natural clusters and is not dependent on any driving objective function
- Hence such a cluster can be used to analyze the portfolio on different target attributes
- For instance, say a decision tree is built on customer profitability in next 3 months. This segmentation cannot be used for making retention strategy for each segment. If segmentation were developed through clustering, both retention and profitability strategy can be built on these segments.

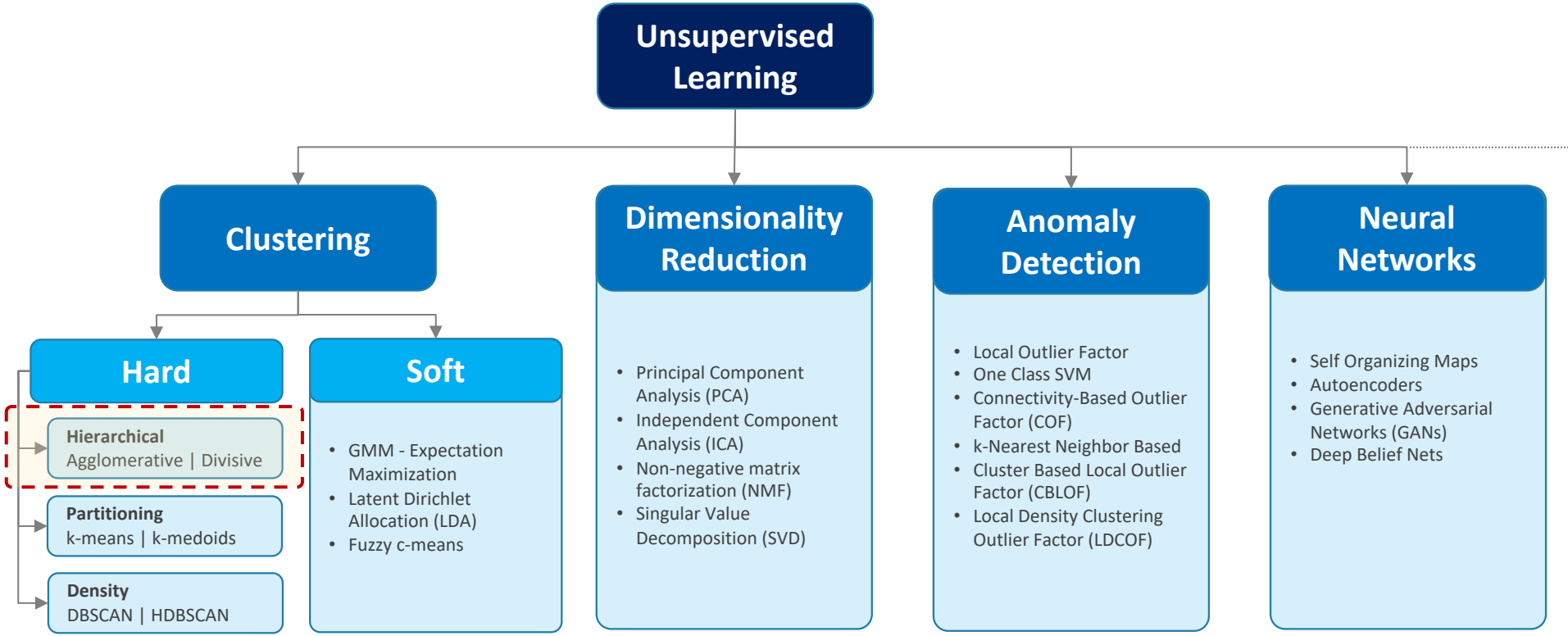
Types: Hard vs Soft Clustering

In **Hard Clustering** each data point either belongs to a cluster completely or not. Example: k-means, k-medoids, DBSCAN



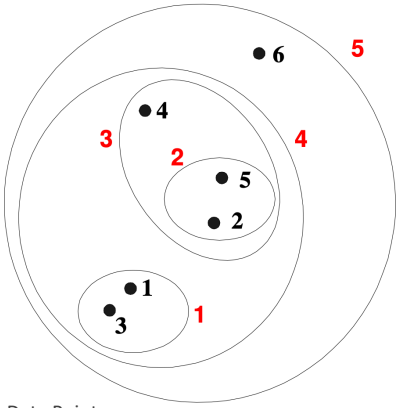
In **Soft clustering,** instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. Example: fuzzy c-means, LDA




Unsupervised Learning
Clustering
Hard
Hierarchical
Agglomerative | Divisive
Partitioning
k-means | k-medoids
Density
DBSCAN | HDBSCAN
Soft
GMM - Expectation Maximization
Latent Dirichlet Allocation (LDA)
Fuzzy c-means
Dimensionality Reduction
Principal Component Analysis (PCA)
Independent Component Analysis (ICA)
Non-negative matrix factorization (NMF)
Singular Value Decomposition (SVD)
Anomaly Detection
Local Outlier Factor
One Class SVM
Connectivity-Based Outlier Factor (COF)
k-Nearest Neighbor Based
Cluster Based Local Outlier Factor (CBLOF)
Local Density Clustering Outlier Factor (LDCOF)
Neural Networks
Self Organizing Maps
Autoencoders
Generative Adversarial Networks (GANs)
Deep Belief Nets

Hierarchical Clustering - Hierarchical decomposition of the data based on group similarities

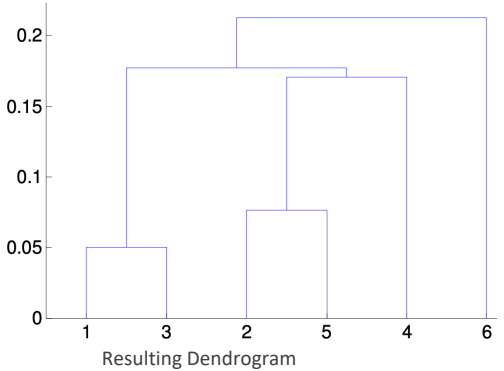
- Hierarchical algorithms produce a nested sequence (**dendrogram**) of clusters, with single all-inclusive cluster at the top and singleton clusters of individual points at the bottom
- The hierarchy can be formed in top-down (divisive) or bottom-up (agglomerative)
- The merging or splitting stops once the desired number of clusters has been formed



Data Points

Agglomerative:

- Start with the points as individual clusters
- At each step, **merge the closest pair of clusters** until only one cluster left
- Merging involves calculating a **dissimilarity** between each merged pair and the other samples



Resulting Dendrogram

Divisive:

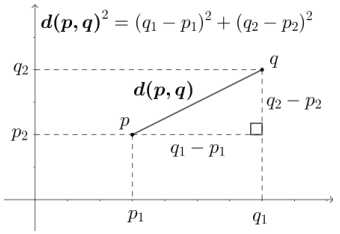
- Start with one, all-inclusive cluster
- At each iteration, we split the farthest point in the cluster
- Repeat this process until each cluster only contains a single point
- Divisive clustering is very rarely used in industry

Dissimilarity = f(**Distance Metric**, Linkage Criterion)

Distance Metrics

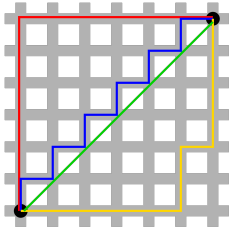
Euclidean Distance: Most used distance metric. Length of the path connecting two points directly. Equivalent to “Pythagorean Theorem”.

$$d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$



Manhattan Distance: Distance between two points is the sum of the absolute differences of their Cartesian coordinates. Manhattan distance is also known as Taxicab Geometry, City Block Distance etc.

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|$$



Mahalanobis Distance: Multi-dimensional generalization of the idea of measuring how many standard deviations away P is from the mean of D. Euclidean distance assumes no-correlation between variables. MD solves this measurement problem, as it measures distances between points, taking covariance metric into consideration.

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}$$

Also read about-

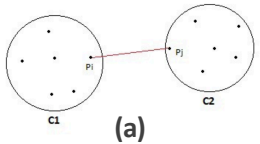
[Cosine Similarity]
[Haversine Distance]
[Hamming Distance]

Dissimilarity = f(Distance Metric, **Linkage Criterion**)

Four common Linkage Criterion

Single-Linkage or Nearest Neighbor: similarity of the closest pair. Generates minimum spanning tree. This can cause premature merging of groups with close pairs, even if those groups are quite dissimilar overall.

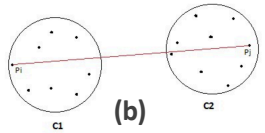
minimum distance $d_{min}(D_i, D_j) = \min_{x \in D_i, y \in D_j} \| x - y \|$



(a)

Complete Linkage or Farthest Point: similarity of the farthest pair. Encourages compact clusters. One drawback is that outliers can cause merging of close groups later than is optimal.

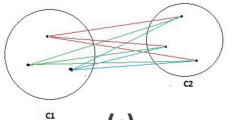
maximum distance $d_{max}(D_i, D_j) = \max_{x \in D_i, y \in D_j} \| x - y \|$



(b)

Group Average: similarity between groups.

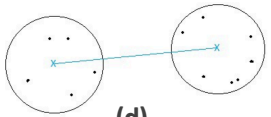
average distance $d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{x \in D_i} \sum_{y \in D_j} \| x - y \|$



(c)

Centroid Similarity: each iteration merges the clusters with the most similar central point.

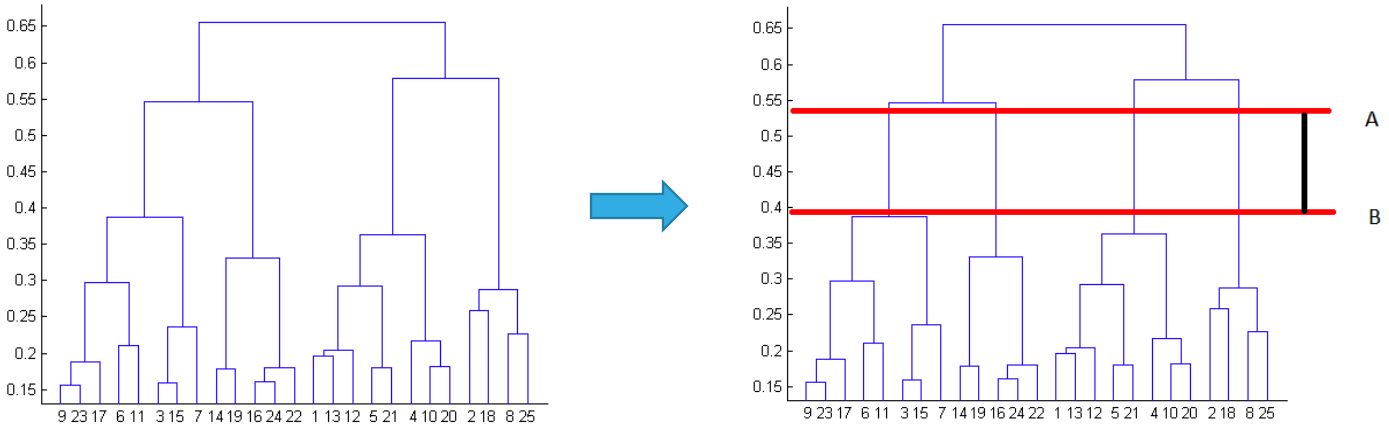
mean distance $d_{mean}(D_i, D_j) = \| \mu_i - \mu_j \|$


(d)

[Read More – SciPy.org Documentation]

Hierarchical Clustering: How do we decide the number of clusters?

- We make use of a concept called a Dendrogram
- A dendrogram is a tree-like diagram that records the sequences of merges or splits
- The vertical line represents the distance between these samples or clusters
- To get clusters we draw a horizontal line by generally cutting the tallest vertical line
- #clusters will be the #vertical-lines being intersected by the line drawn at threshold



Hierarchical Clustering: Advantages and Disadvantages

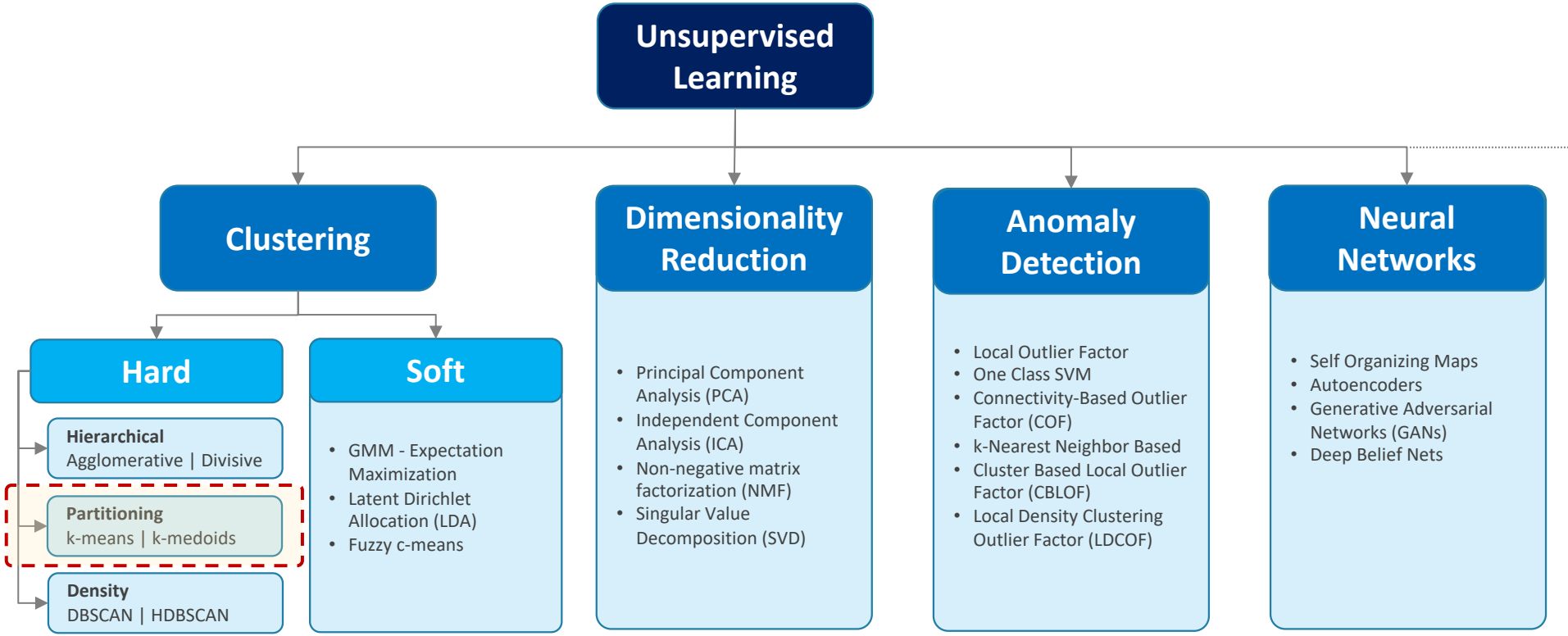
Advantages

- **Do not have to assume any particular number of clusters**
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

- **They may correspond to meaningful taxonomies**
  - Ex: In biological sciences (animal kingdom, phylogeny reconstruction)
  - Ex: In Retail (Item Hierarchies)

Disadvantages

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- **Time Complexity $O(N^3)$**
  - There are N steps and at each step the size, N^2, proximity matrix must be updated and searched
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches


Unsupervised Learning
Clustering
Hard
Hierarchical
Agglomerative | Divisive
Partitioning
k-means | k-medoids
Density
DBSCAN | HDBSCAN
Soft
• GMM - Expectation Maximization
• Latent Dirichlet Allocation (LDA)
• Fuzzy c-means
Dimensionality Reduction
• Principal Component Analysis (PCA)
• Independent Component Analysis (ICA)
• Non-negative matrix factorization (NMF)
• Singular Value Decomposition (SVD)
Anomaly Detection
• Local Outlier Factor
• One Class SVM
• Connectivity-Based Outlier Factor (COF)
• k-Nearest Neighbor Based
• Cluster Based Local Outlier Factor (CBLOF)
• Local Density Clustering Outlier Factor (LDCOF)
Neural Networks
• Self Organizing Maps
• Autoencoders
• Generative Adversarial Networks (GANs)
• Deep Belief Nets

Partitioning Based Clustering

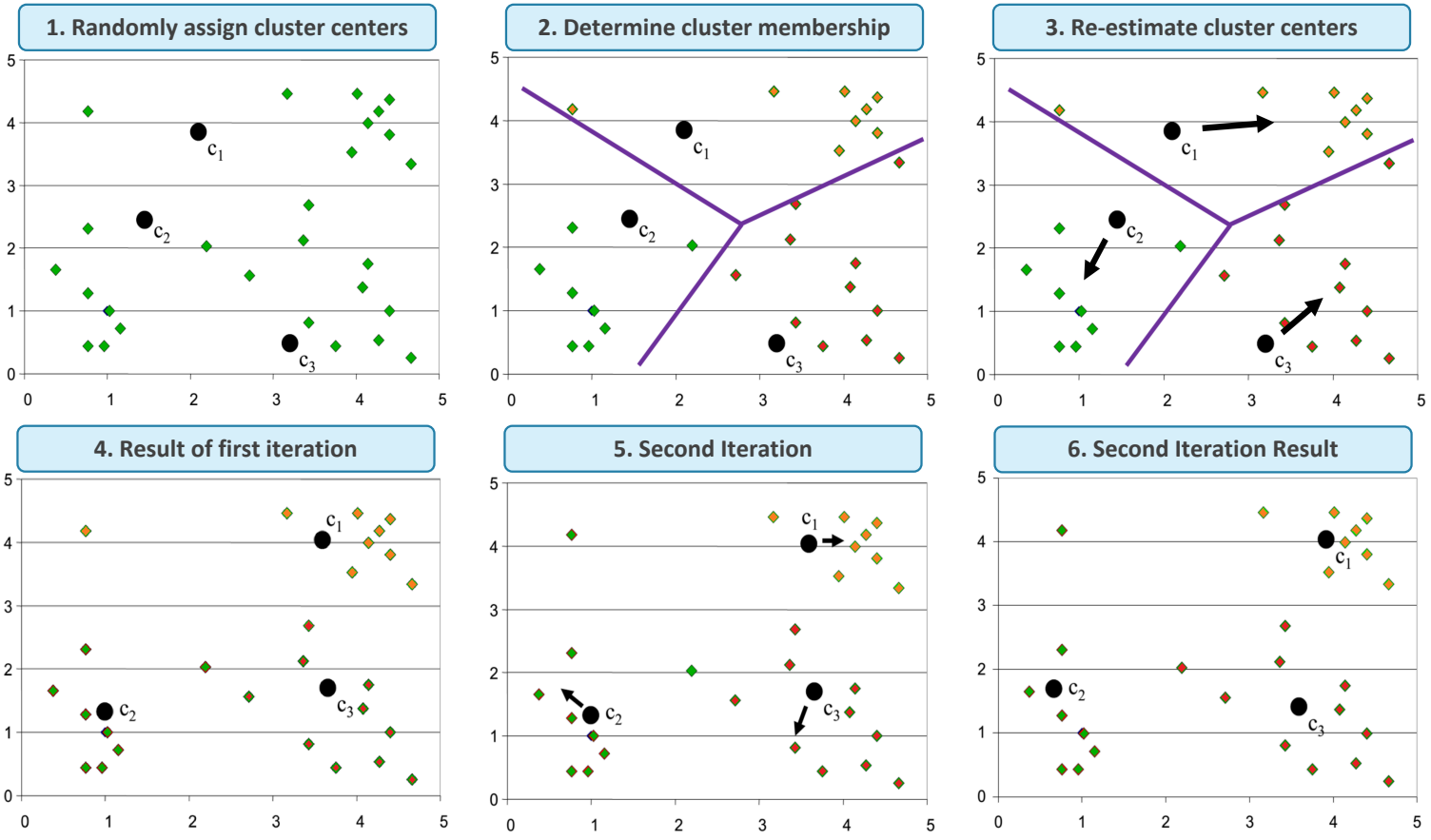
- Decompose a set of N objects into k clusters such that the partitions **optimize a certain criterion function**
- Notion of similarity (dissimilarity) is derived by **closeness of a data point to the center of gravity (centroid)**
- **Iterative** clustering algorithms
- **Number of clusters (k) required** at the end have to be mentioned beforehand
- **Important to have prior knowledge** of the dataset
- k-means, k-medoids, CLARANS

k-means Algorithm (Most popular clustering method)

Algorithm Steps

1. Choose the number of clusters, k
2. Randomly generate k clusters and determine the cluster centers, or directly generate k random points as cluster centers
3. Assign each point to the nearest cluster center
4. Recompute the new cluster centers
5. Repeat the two previous steps until some convergence criterion is met (usually that the assignment hasn't changed)

k-means example (This is boring!)



[Interesting Visualization Here!]

k-means convergence (stopping) criterion

- no (or minimum) re-assignments of data points to different clusters, or
- no (or minimum) change of centroids, or
- minimum decrease in the sum of squared error (SSE)

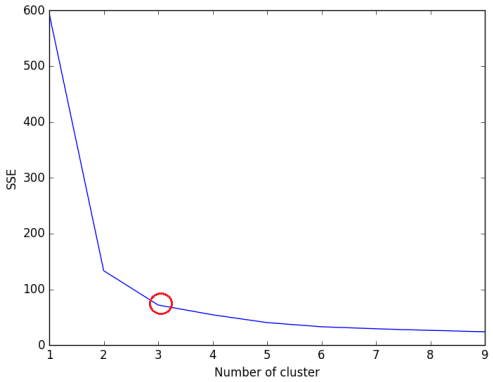
$$\text{SSE} = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} d(\mathbf{x}, \mathbf{m}_j)^2$$

where

- C_j is the j^{th} cluster
- m_j is the centroid of cluster C_j (the mean vector of all the data points in C_j)
- $d(x, m_j)$ is the distance between data point x and centroid mj

Find k (#clusters)?

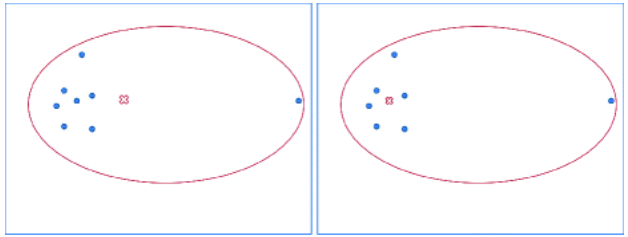
- Plot distortion (SSE) on y-axis for different values of k (number of clusters)
- Choose k (number of clusters) so that adding another cluster doesn't give much better modeling of the data.
- Elbow is created at some point as the marginal gain in explained variance will drop, giving an angle in the graph



Also read about – Silhouette Coefficient

k-medoids (PAM)

- In k-medoids or Partitioning Around Medoids (PAM) method a cluster is represented by its medoid
- A medoid can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal, that is, it is a most centrally located point in the cluster
- Medoids are more resistant to outliers and noise compared to centroids as it minimizes a sum of pairwise dissimilarities instead of a sum of squared (euclidean) distances



(a) Mean (b) Medoid

Algorithm Steps

- PAM begins by selecting randomly a data point as medoid for each of the k clusters
- Then, each of the non-selected objects is grouped with the medoid to which it is the most similar
- PAM then iteratively replaces one of the medoids by one of the non-medoids objects yielding the greatest improvement in the cost function

Partitioning Clustering: Advantages and Disadvantages

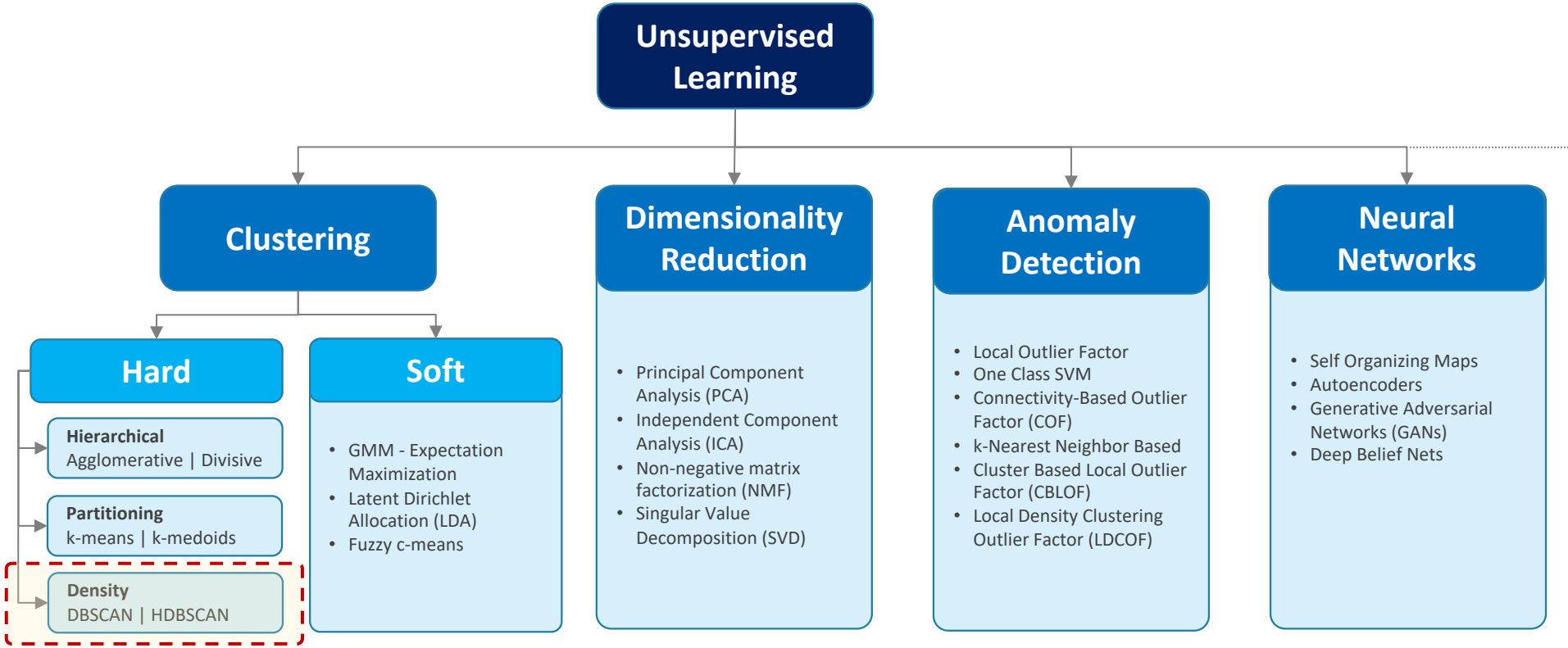
Advantages

- Simple: Easy to understand and to implement
- Efficient: Time complexity: O(tkn), where n is #data-points, k is #clusters and t is #iterations
  - Since both k and t are small. k-means is considered a linear algorithm
- Suitable for datasets with compact spherical clusters that are well-separated

Let's confirm these pointers!!

Disadvantages

- Reliance on the user to specify the number of clusters in advance
- High sensitivity to initialization phase, noise and outliers
- Inability to deal with non-convex clusters of varying size and density
- Severe effectiveness degradation in high dimensional spaces as almost all pairs of points are about as far away as average; the concept of distance between points in high dimensional spaces is ill-defined


Unsupervised Learning
Clustering
Hard
Hierarchical
Agglomerative | Divisive
Partitioning
k-means | k-medoids
Density
DBSCAN | HDBSCAN
Soft
GMM - Expectation Maximization
Latent Dirichlet Allocation (LDA)
Fuzzy c-means
Dimensionality Reduction
Principal Component Analysis (PCA)
Independent Component Analysis (ICA)
Non-negative matrix factorization (NMF)
Singular Value Decomposition (SVD)
Anomaly Detection
Local Outlier Factor
One Class SVM
Connectivity-Based Outlier Factor (COF)
k-Nearest Neighbor Based
Cluster Based Local Outlier Factor (CBLOF)
Local Density Clustering Outlier Factor (LDCOF)
Neural Networks
Self Organizing Maps
Autoencoders
Generative Adversarial Networks (GANs)
Deep Belief Nets

Density Based Clustering

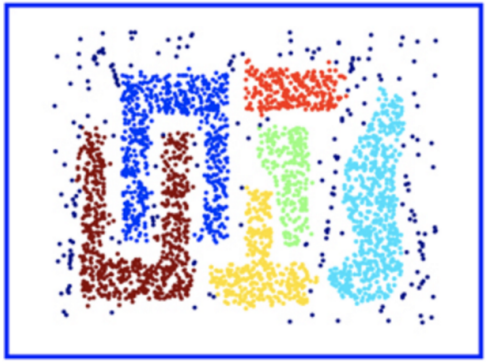
- Density-based clustering methods group neighboring objects into clusters based on local density (density regions) conditions rather than proximity between objects

- These methods regard clusters as dense regions being separated by low density noisy regions. (Helps identify outliers)

- Density-based methods have noise tolerance, and can discover non-convex clusters

- Example- DBSCAN, HDBSCAN, OPTICS

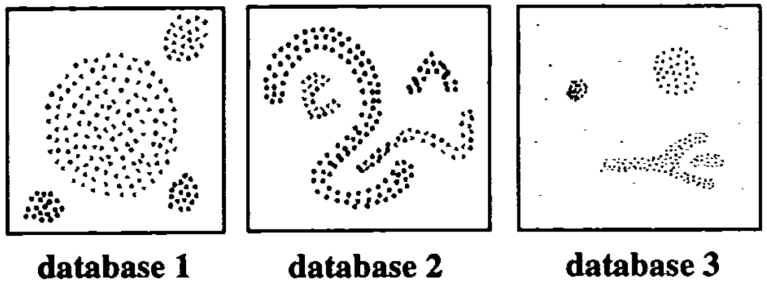
Non-Convex Shaped Clusters



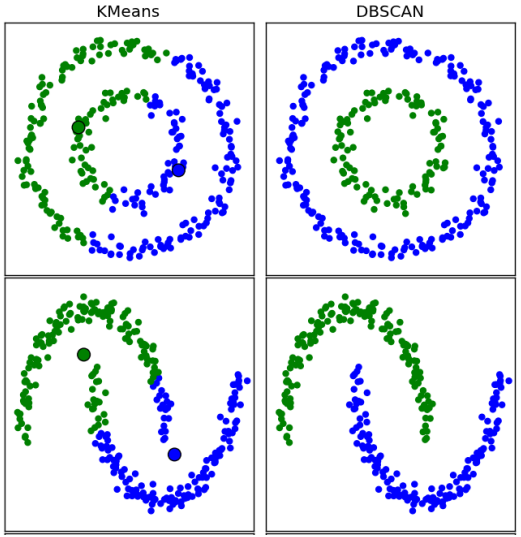
Density Based Clustering Output

DBSCAN – If possible, read complete paper. Very interesting.

- **DBSCAN**- density-based spatial clustering of applications with noise
- First introduced in 1996 by ester et. Al.
- Test of Time Award Winner at SIKDD-2014



Clusters identified by DBSCAN
Source: DBSCAN Paper



DBSCAN Algorithm

The algorithm has two parameters:

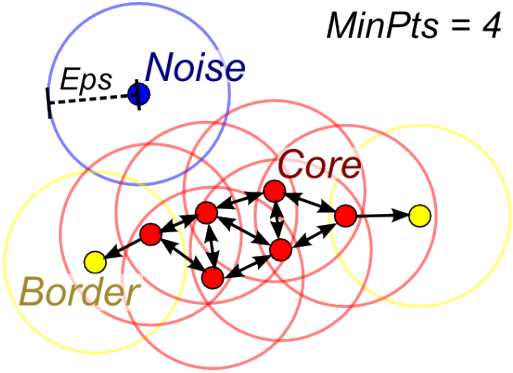
- **ε:** The radius of our neighborhoods around a data point p
- **minPts:** The minimum number of data points we want in a neighborhood to define a cluster

Using these two parameters, DBSCAN categories data points into 3 categories:

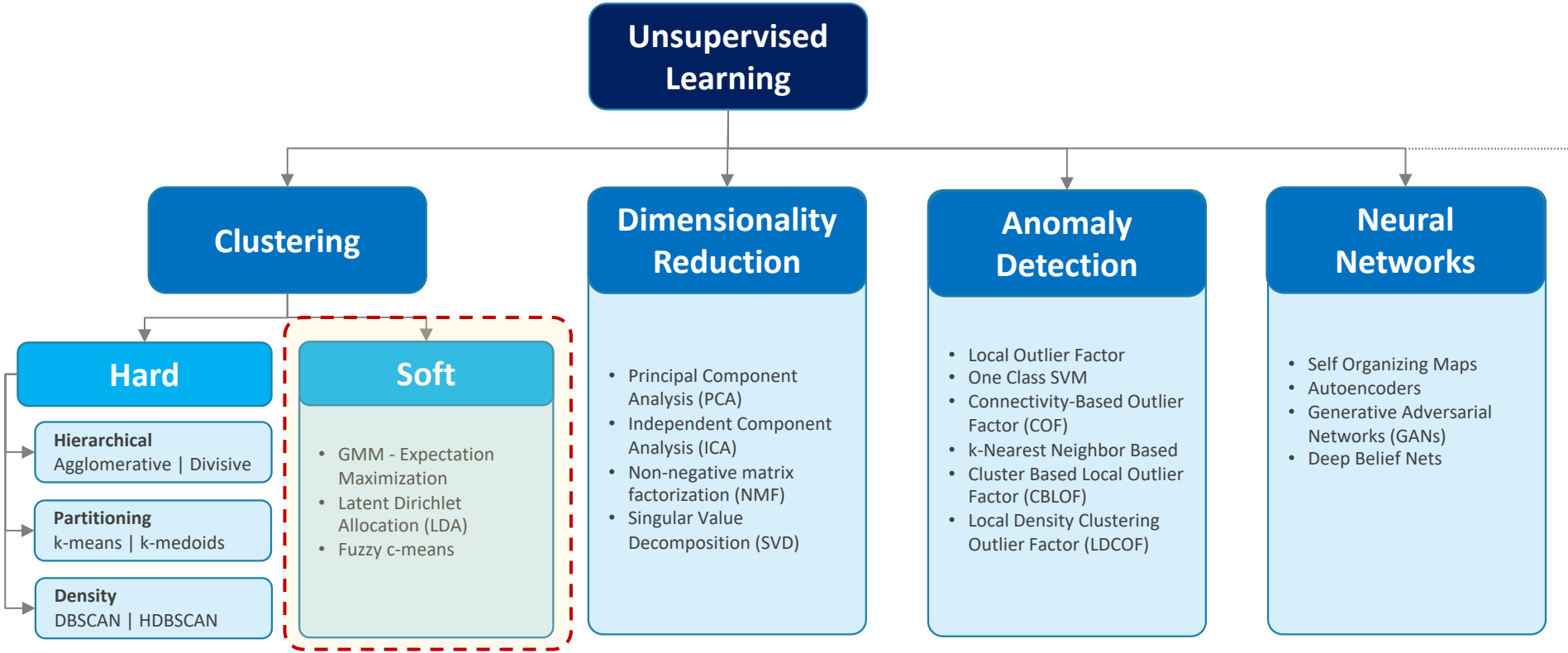
- **Core Points:** A data point p is a core point if Nbhd(p,ε) [ε-neighborhood of p] contains at least minPts ; |Nbhd(p,ε)| >= minPts

- **Border Points:** A data point q is a border point if Nbhd(q, ε) contains less than minPts data points, but q is reachable from some core point p

- **Outlier:** A data point o is an outlier if it is neither a core point nor a border point. Essentially, this is the “other” class



[Visualizing DBSCAN Clustering]

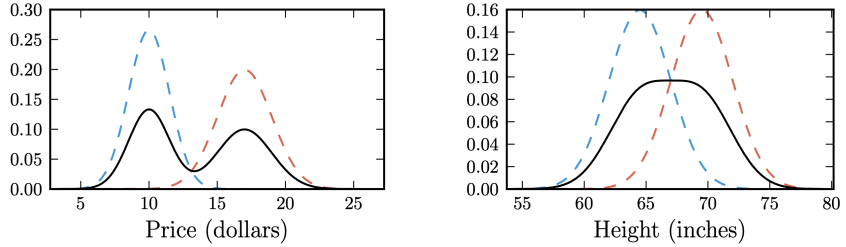

Unsupervised Learning
Clustering
Hard
Hierarchical
Agglomerative | Divisive
Partitioning
k-means | k-medoids
Density
DBSCAN | HDBSCAN
Soft
GMM - Expectation Maximization
Latent Dirichlet Allocation (LDA)
Fuzzy c-means
Dimensionality Reduction
Principal Component Analysis (PCA)
Independent Component Analysis (ICA)
Non-negative matrix factorization (NMF)
Singular Value Decomposition (SVD)
Anomaly Detection
Local Outlier Factor
One Class SVM
Connectivity-Based Outlier Factor (COF)
k-Nearest Neighbor Based
Cluster Based Local Outlier Factor (CBLOF)
Local Density Clustering Outlier Factor (LDCOF)
Neural Networks
Self Organizing Maps
Autoencoders
Generative Adversarial Networks (GANs)
Deep Belief Nets

Gaussian Mixture Models

- A Gaussian Mixture Model (GMM) is useful for modeling data that comes from one of several groups: the groups might be different from each other, but data points within the same group can be well-modeled by a Gaussian distribution.
- To cluster the observations we do reverse engineering and identify the probabilities of each Gaussian distribution for a given observation.

Ex: Suppose the price of randomly chosen paperback book is normally distributed with mean $10.00 and Std.Dev $1.00. Similarly, the price of a randomly chosen hardback is normally distributed with mean $17 and Std.Dev $1.50. Is the price of a randomly chosen book normally distributed? The answer is no.

We can see this by looking at the fundamental property of the normal distribution: it's highest near the center, and quickly drops off farther away. But, the distribution of a randomly chosen book is bimodal: the center of the distribution is near $13, but the probability of finding a book near that price is lower than the probability of finding a book for a few dollars more or a few dollars less.



(a) Probability density for paperback books (blue), hardback books (red), and all books (black, solid)

(b) Probability density for heights of women (blue), heights of men (red), and all heights (black, solid)

Figure 1: Two Gaussian mixture models: the component densities (which are Gaussian) are shown in dotted red and blue lines, while the overall density (which is not) is shown as a solid black line.

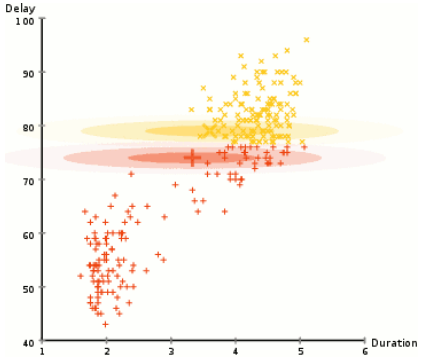
GMM using Expectation Maximization

- There are different ways to learn model parameters of GMM. Expectation Maximization is one of the effective technique

Expectation Maximization for mixture models consists of two steps.

- The first step, known as **the expectation step or E step**, consists of calculating the expectation (say probability) of the component (cluster) assignments C_k for given model parameters (means, variances, component weights)
- The second step is known as **the maximization step or M step**, which consists of maximizing the expectations calculated in the E step with respect to the model parameters. This step consists of updating the model parameters.
- The entire iterative process repeats until the algorithm converges, giving a maximum likelihood estimate.

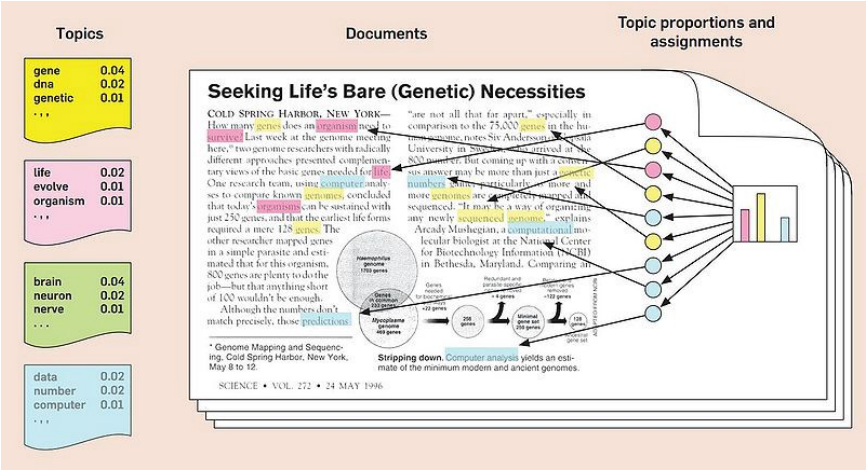
Intuitively, the algorithm works because knowing the component assignment for each data point makes solving for model parameters easy, while knowing model parameters makes inferring component assignment easy



Latent Dirichlet Allocation (LDA)

- Topic modeling is an unsupervised problem which requires statistical modeling for discovering the abstract “topics” that occur in a collection of documents
- Latent Dirichlet Allocation (LDA) is an example of topic model and is used to classify text in a document to topics
- It builds two models, modeled as Dirichlet distributions 1) topics per document model 2) words per topic model
- LDA allows for ‘fuzzy’ memberships. This provides a more nuanced way of recommending similar items, finding duplicates, or discovering user profiles/personas.
- If you choose the number of topics to be less than the documents, using LDA is a way of reducing the dimensionality (the number of rows and columns) of the original composite versus part data set

[Read a very detailed and intuitive blog here!!]



Self Read - Techniques

Fuzzy C-Means Clustering - Link

- FCM is a method of clustering which allows one piece of data to belong to two or more clusters (soft clustering)
- Differs from the k-means objective function by the addition of the membership values (degree to which data point belongs to a cluster) and the fuzzifier (determines the level of cluster fuzziness)

HDBSCAN – Link

- HDBSCAN extends DBSCAN by converting it into a hierarchical clustering algorithm, and then using a technique to extract a flat clustering based in the stability of clusters
- Key advantage is that HDBSCAN takes care of varying density across feature-space as compared to global density parameters in DBSCAN

Industry Use Cases

Detect Natural Patterns like partitioning customers into segments for better recommendation/targeting

- Recommendation engines
- Market segmentation
- Social network analysis
- Search result grouping
- Automated IT Alerts
- Medical imaging
- Image segmentation

Preprocessing Step For Supervised Learning/Semi-Supervised- Clustering the data gives cluster labels as another feature for Supervised Models. Also, in Semi-Supervised cluster labels could be used as dependent variable

Reduce Data Size- Instead of dealing with complete massive dataset, cluster them first, before analysis. Soft clustering techniques could give you probability vectors and thus help reduce dimensionality. Example- **LDA**

Anomaly Detection- Many clustering techniques help detect noise/outliers in the data and could be used to develop anomaly detection systems. DBSCAN, GMM based Fraud Detection

Closing Remarks – Start Learning By Doing

- Before choosing a clustering technique look for business understanding around the natural clusters supposed to be created
- Choose an algorithm that is aligned with the requirements. For example- In e-commerce a same shoe could be displayed in both Sports Shoes vs Casual Shoes category. Hence, you need soft clustering here and not hard clustering
- Clustering is quite an old stream and still is in active research. More algorithms are coming every year
- Clustering is hard to evaluate because of absence of ground truth, but very useful in practice
- Clustering is highly application dependent (and to some extent subjective)

Sources

- https://www.technologyreview.com/s/612437/what-is-machine-learning-we-drew-you-another-flowchart/
- https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/
- http://www.mit.edu/~9.54/fall14/slides/Class13.pdf
- https://www.geeksforgeeks.org/different-types-clustering-algorithm/
- https://pdfs.semanticscholar.org/68d8/a39bc9035fef6ec504cbd04e900cadd504aa.pdf
- https://www.statisticshowto.datasciencecentral.com/hierarchical-clustering/
- https://blog.dominodatalab.com/topology-and-density-based-clustering/
- https://brilliant.org/wiki/gaussian-mixture-model/
- https://pythonprogramminglanguage.com/kmeans-text-clustering/

Thank You!
Questions Please..

Contact: Eshan Jain
Senior Data Scientist @ Walmart Labs
eshanjain.ej+greyatom@gmail.com
Linkedin: https://www.linkedin.com/in/eshanjain/