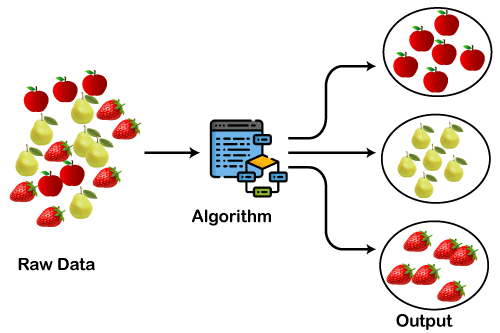
CLUSTERING

* Clustering can be achieved in the unsupervised learning method.
* Clustering defined as, **way of grouping** the data points into **different clusters**, consisting of similar data points. The objects with the **possible similarities remain in a group** **that has less or no similarities with another group**.



KMEANS:

* K-Means Clustering is an **Unsupervised Learning algorithm**, which **groups** **the** **unlabelled dataset into different clusters**.
* Here **K defines the number of pre-defined clusters** that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
* It is **an iterative process of assigning each data point to the groups and slowly data points get clustered based on similar features**. The objective is to minimize the sum of distances between the data points and the cluster centroid, to identify the correct group each data point should belong to.



Pros:

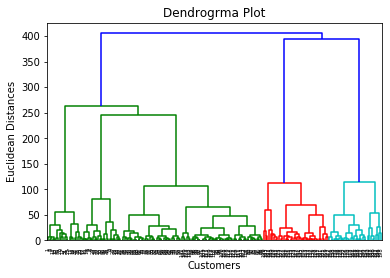
* High Performance
* Easy to Use
* Unlabelled Data
* Result Interpretation

Cons:

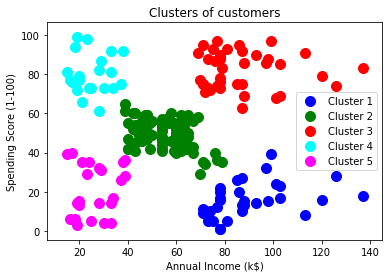
* Result Repeatability
* To much of Manual Effort Required
* Spherical Clustering Only
* Clusters Everything

AGGLOMERATIVE:

* Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
* we will find the optimal number of clusters using the Dendrogram for our model.



* **n\_clusters=5**: It defines the number of clusters, and we have taken here 5 because it is the optimal number of clusters.



Pros:

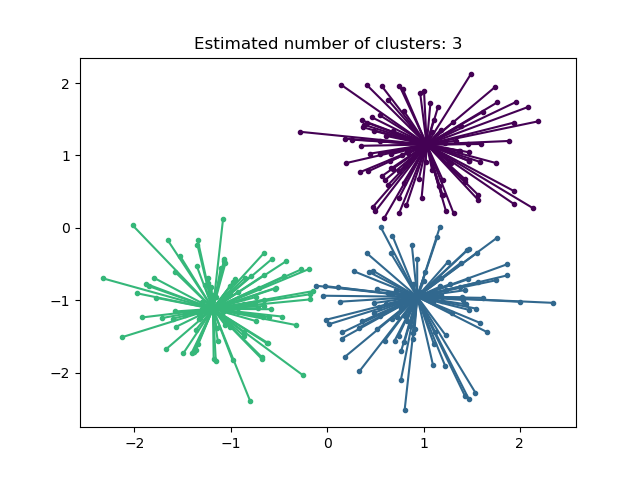
* We can obtain the optimal number of clusters from the model itself, human intervention not required.
* Dendrograms help us in clear visualization, which is practical and easy to understand.

Cons:

* Not suitable for large datasets due to high time and space complexity.
* There is no mathematical objective for Hierarchical clustering.
* All the approaches to calculate the similarity between clusters has their own disadvantages.

AFFINITY PROPOGATION:

* Affinity Propagation, instead, takes as input measures of similarity between pairs of data points, and simultaneously considers all data points as potential exemplars.
* Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges



* The inventors of affinity propagation showed it is **better for certain** **computer vision and computational biology tasks,** e.g. clustering of pictures of human faces and identifying regulated transcripts, than k-means, even when k-means was allowed many random restarts and initialized using PCA.

Pros:

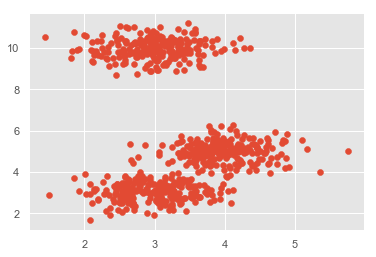
* The user doesn’t need to specify the number of clusters (but does need to specify ‘sample preference’ and ‘damping’ hyperparameters).

Cons:

* The main disadvantage of Affinity Propagation is that it’s quite slow and memory-heavy, making it difficult to scale to larger datasets.

MEANSHIFT:

* Mean-shift algorithm basically assigns the datapoints to the clusters iteratively by shifting points towards the highest density of datapoints i.e. cluster centroid.
* The difference between K-Means algorithm and Mean-Shift is that later one does not need to specify the number of clusters in advance because the number of clusters will be determined by the algorithm w.r.t data.



Pros:

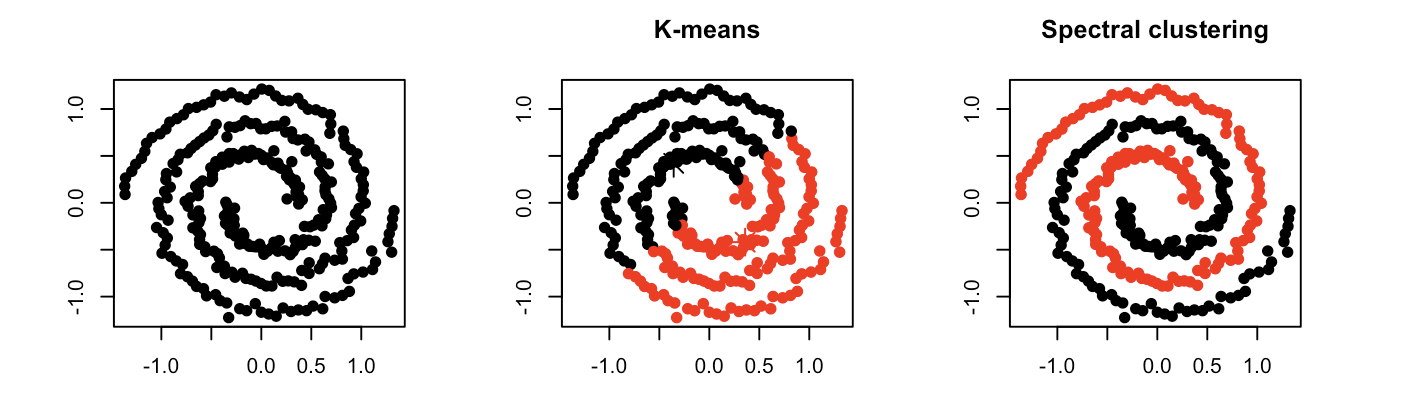
* It does not need to make any model assumption as like in K-means or Gaussian mixture.
* It can also model the complex clusters which have nonconvex shape.
* It only needs one parameter named bandwidth which automatically determines the number of clusters.
* No problem generated from outliers.

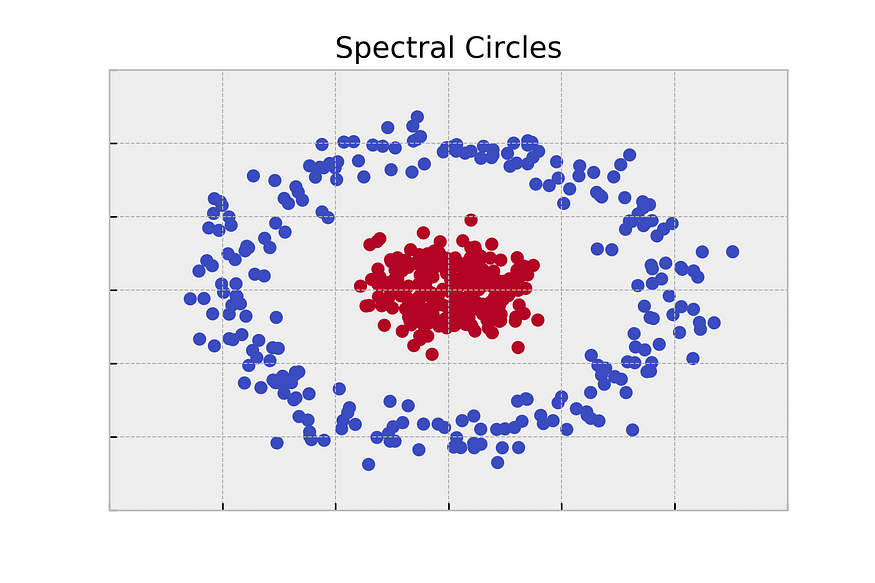
Cons:

* We do not have any direct control on the number of clusters but in some applications, we need a specific number of clusters.
* It cannot differentiate between meaningful and meaningless modes.

SPECTRAL CLUSTERING:

* Thus, spectral clustering is a **graph partitioning problem**.
* **The nodes are then mapped** to a low-dimensional space that can be **easily segregated to form clusters**. No assumption is made about the shape/form of the clusters.
* The goal of spectral clustering is to **cluster data that is connected** but **not necessarily compact or clustered** within convex boundaries.





Pros:

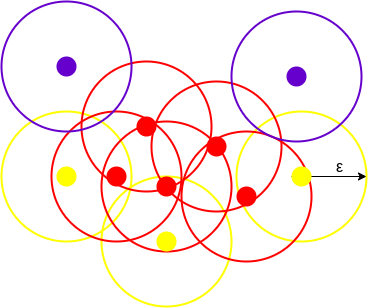
* practically work well even some assumptions are broken.
* simple, easy to implement.
* easy to interpret the clustering results.
* fast and efficient in terms of computational cost.

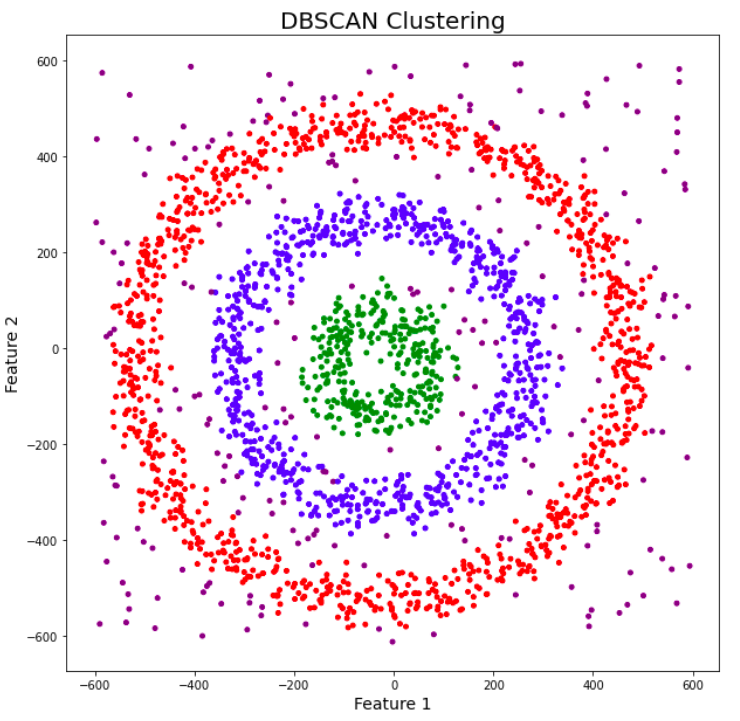
Cons:

* uniform effect: often produce clusters with relatively uniform size even if the input data have different cluster size
* spherical assumption hard to satisfied: correlation between features break it, would put extra weights on correlated features
* cannot find non-convex clusters or clusters with unusual shapes
* different densities: may work poorly with clusters with different densities but spherical shape
* Sensitive to Outliers.

DBSCAN:

* **DBSCAN** stands for **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise.
* It can identify clusters in **large spatial datasets** by looking at the local density of the data points.
* **The most exciting feature of DBSCAN clustering is that it is robust to outliers**.
* DBSCAN creates a circle of epsilon radius around every data point and classifies them into **Core** point, **Border** point, and **Noise**.





Pros:

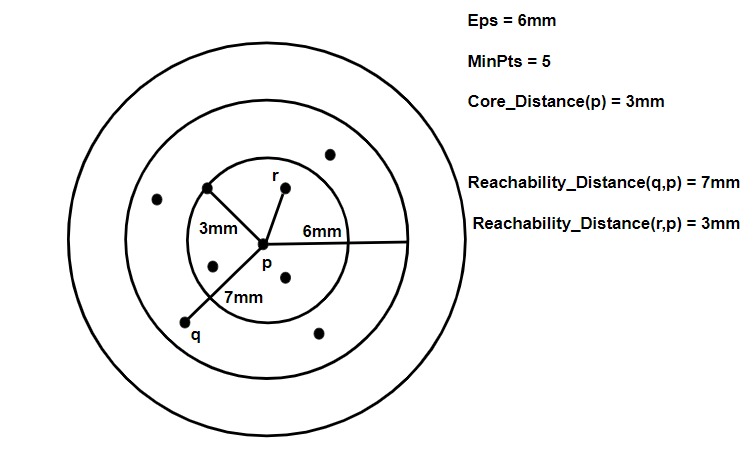
* **Handles irregularly shaped and sized clusters.**
* **Robust to outliers.**
* **Does not require the number of clusters to be specified.**
* **Less sensitive to initialization conditions.**
* **Relatively fast.**

Cons:

* **Difficult to incorporate categorical features**
* **Requires a drop in density to detect cluster borders**
* **Struggles with clusters of varying density**
* **Struggles with high dimensional data**

ORBITS:

* **Ordering Points To Identify Cluster Structure** (OPTICS) ) is a density-based clustering technique that allows partitioning data into groups with similar characteristics.
* Its addresses one of the DBSCAN’s major weaknesses. The problem of **detecting meaningful clusters** in data of varying density.
* In a density-based clustering, clusters are defined as dense regions of data points separated by low-density regions.
* It adds two more terms to the concepts of DBSCAN clustering. They are: **Core Distance and Reachability Distance**.



Pros:

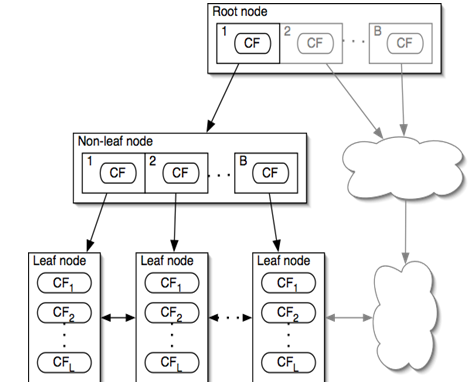
* OPTICS clustering **doesn’t require a predefined number of clusters** in advance.
* Clusters can be of **any shape, including non-spherical ones**.
* Able to identify **outliers** (noise data)

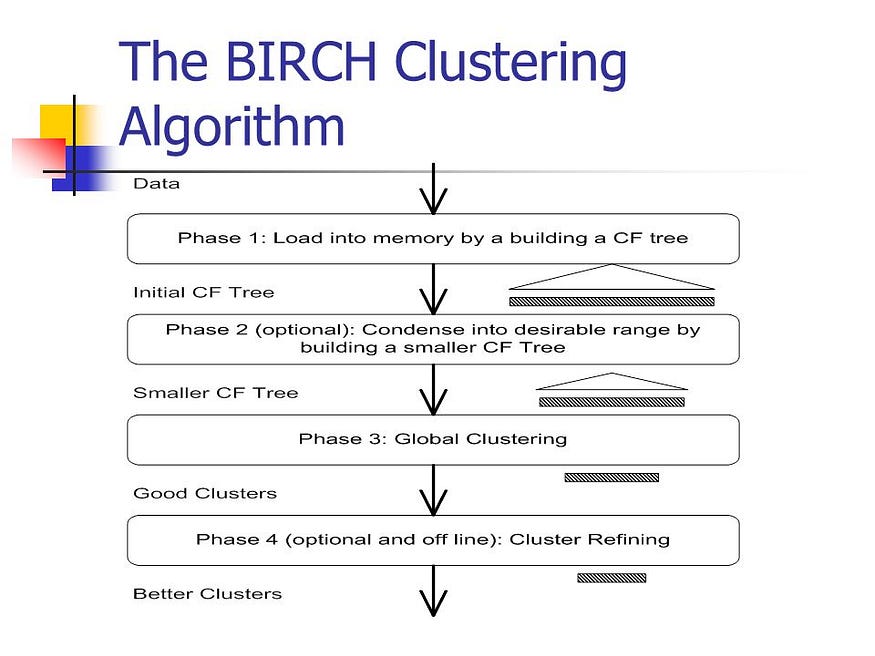
Cons:

* It fails if there are **no density drops between clusters**.
* It is also sensitive to parameters that define density( radius and the minimum number of points) and **proper parameter settings require domain knowledge**.

BIRCH:

* BIRCH defined as **Balanced Iterative Reducing and Clustering hierarchies.**
* BIRCH summarizes **large datasets into smaller, dense regions** called **Clustering** **Feature** (CF) entries.
* It is possible for a CF entry to be composed of other CF entries. Optionally, we can condense this initial CF tree into a smaller CF.
* **Global Clustering**: Applies an existing clustering algorithm on the leaves of the CF tree. A CF tree is a tree where each leaf node contains a sub-cluster. Every entry in a CF tree contains a pointer to a child node and a CF entry made up of the sum of CF entries in the child nodes. Optionally, we can refine these clusters.





Pros:

* Finds a good clustering with a **single scan and improves the quality with a few additional scans.**

Cons:

* works well only for **spherical shape clusters and numeric attributes**.