Unsupervised Machine Learning.

→ The process of making groups of similar data points.

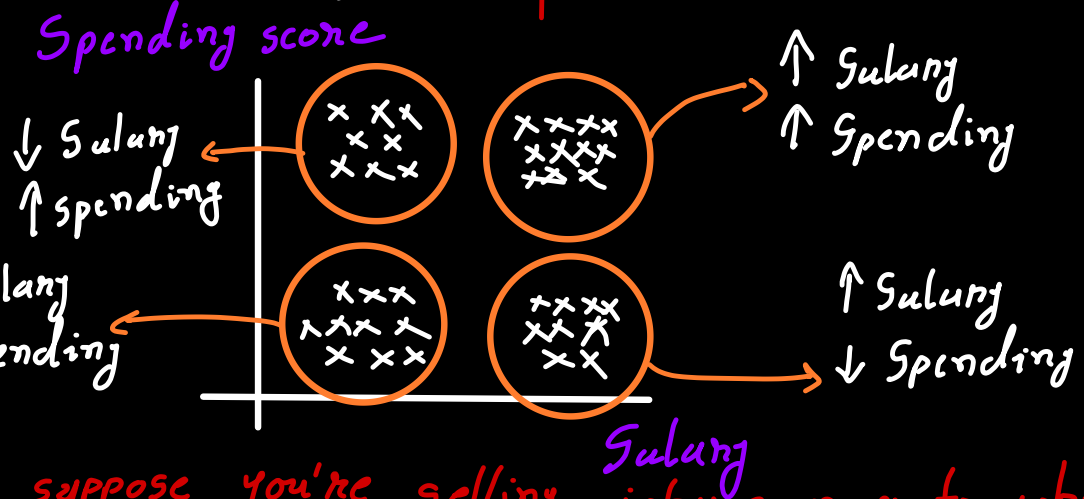
Clustering algorithms

1. K-means clustering
2. Hierarchical clustering.
3. DBScan clustering.

Example

Dataset

Salary	Spending Score
x_1	y_1
x_2	y_2
x_3	y_3

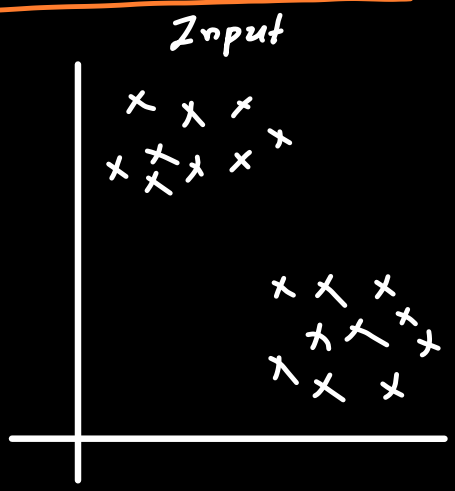


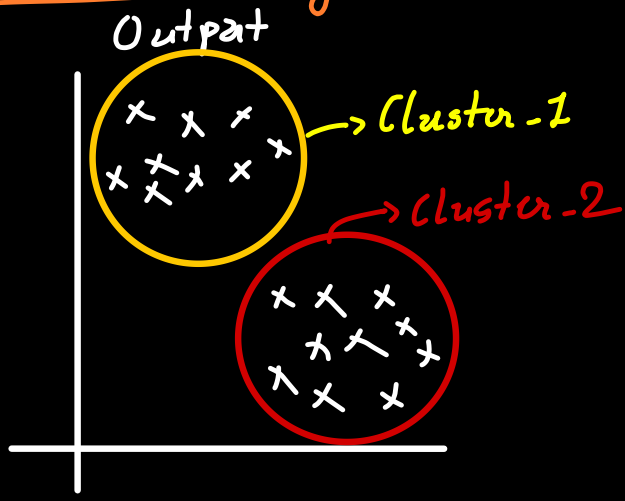
Q. Now suppose you're selling iphone, now to which group, you will give discount....?

→ Probably, you can give discount, whose spending score is high.

→ People with lower spending score may also buy your iphone, if you provide more discount to them.

Geometric intuition of k-means clustering



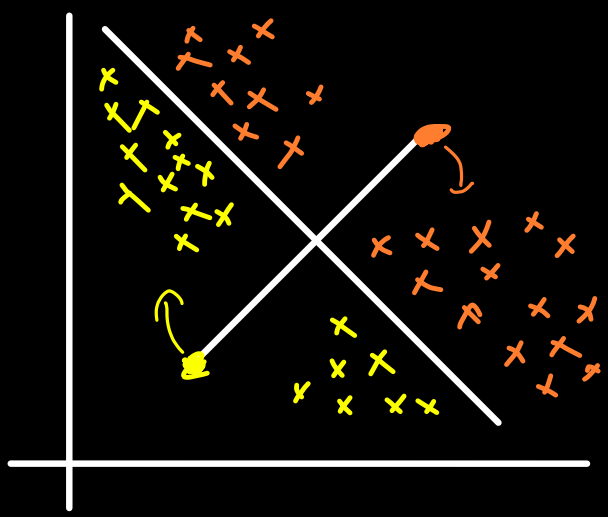


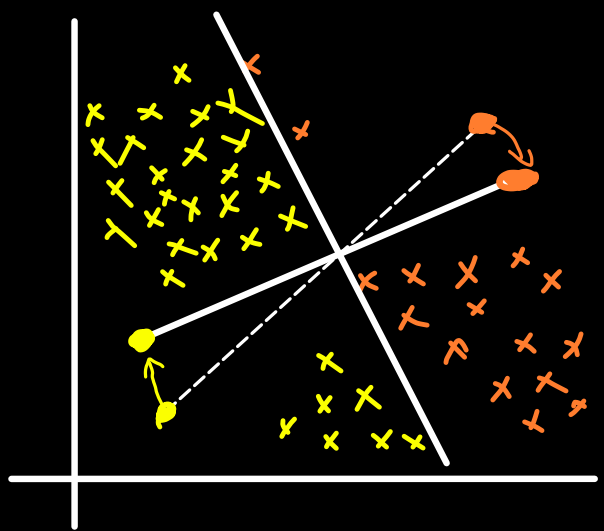
Mathematical Intuition

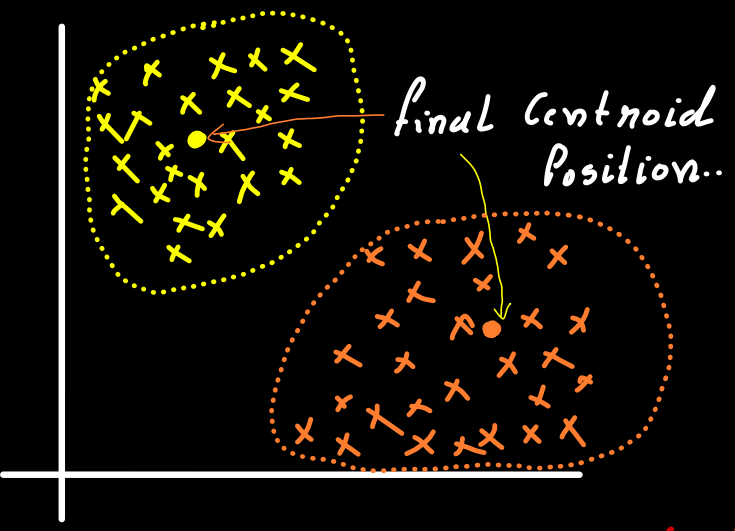
Steps

1. Initialize some k → centroids (Center points of cluster)
2. Points are nearest to the centroid → group them.
3. Move the centroid by calculating mean.

↻ Repeat 2 & 3 step until no major changes are visible in centroid.....







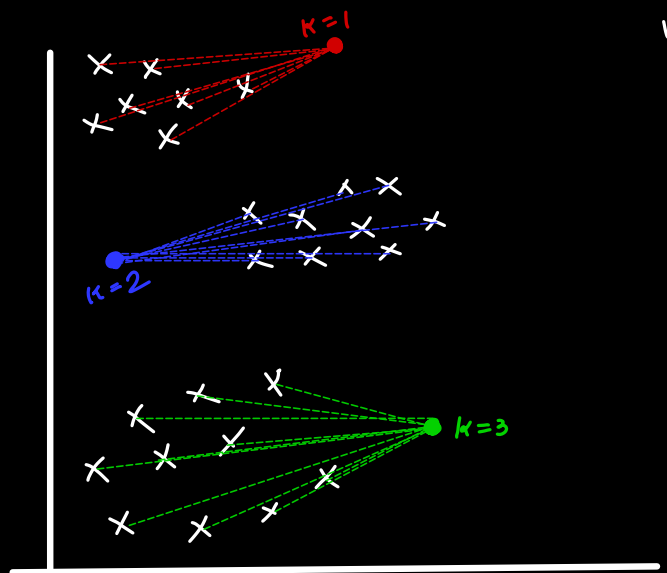
d = distance between point to the nearest centroid

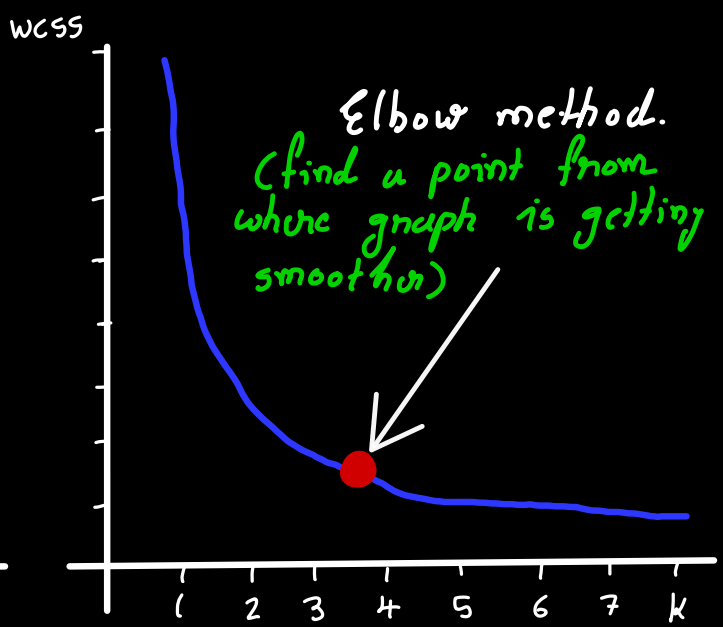
$$WCSS = \sum_{i=1}^{k} (d)^2$$

Q. How do we select the 'k' value...?

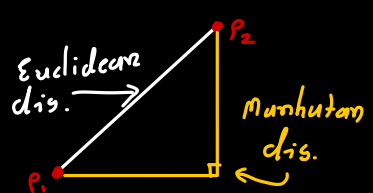
WCSS : Within Cluster Sum of Squares.

→ Initialize k = 1 to 20





→ The distance between points & centroid will be calculated by 2 methods. 1. Euclidean distance
2. Manhatan distance



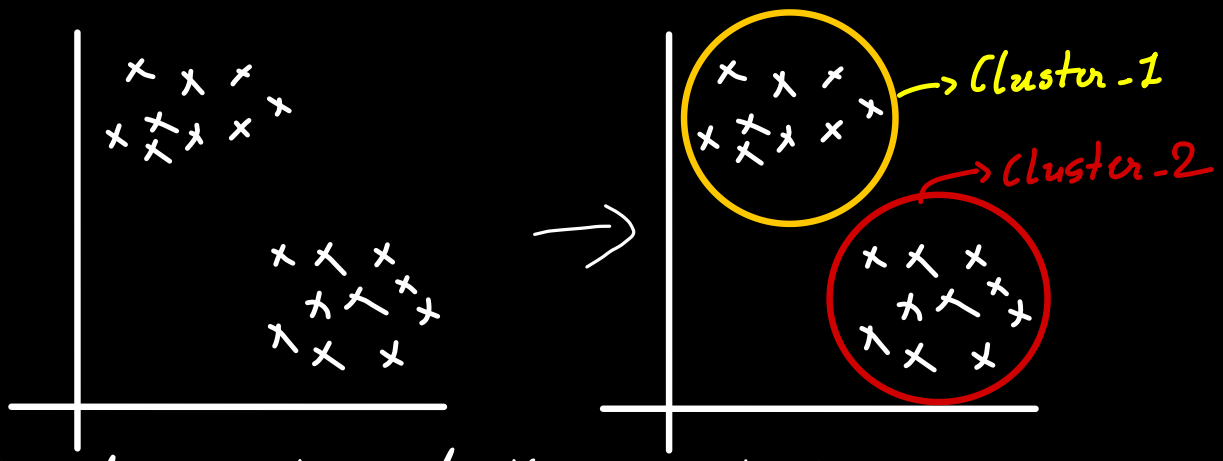
Euclidean distance $= \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

Manhatan distance $= |x_2 - x_1| + |y_2 - y_1|$

Random initialization trap (K-means++)

→ When centroids are located in very very close distance will cause random initialization trap
→ Hence we uses K-means++ initialization technique.
→ This technique initializes centroid on safe distance. (far from each other).

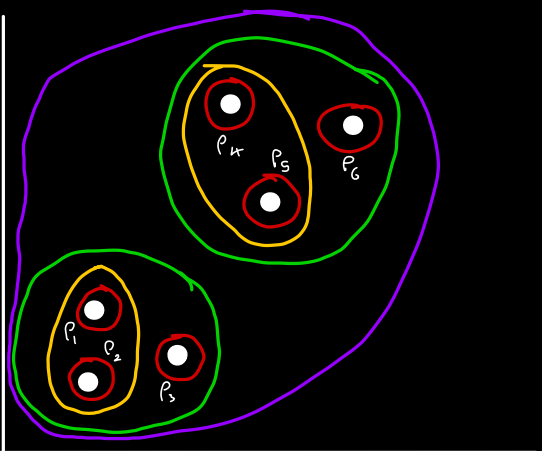
Hierarchical Clustering



→ Hierarchical clustering doesn't uses centroid.

Types

- Agglomerative clustering (Combining)
- Divisive Clustering (Distributing) — Reverse process of AC



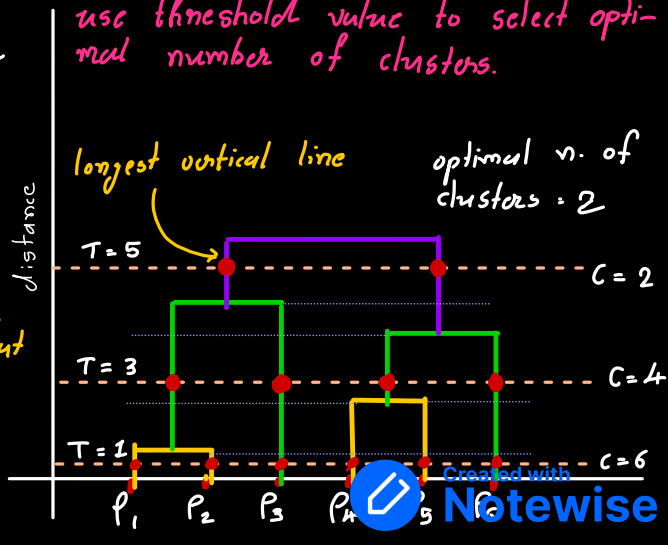
AC Steps

1. At initially, each points will be consider as independent cluster.
2. Find the nearest point and create a new cluster.
3. Repeat step-2 untill there is only one cluster

Q. How many clusters...?
→ Dendogram.

Q. How to select best threshold value?
→ Select a longest vertical line such that no horizontal line passes through it.

use threshold value to select optimal number of clusters.

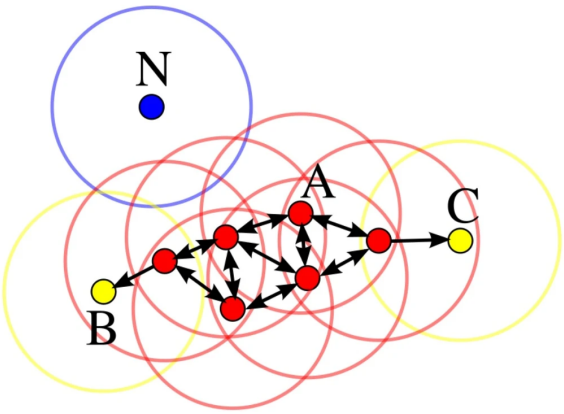


K-means VS Hierarchical clustering.

Scalability & flexibility

1. Dataset size
   - Huge → K-means
   - Small → Hierarchical
2. Type of data
   - Numerical data → Both
   - Variety of data → Hierarchical

DBSCAN CLUSTERING



In this diagram, minPts = 4. Point A and the other red points are core points, because the area surrounding these points in an ε radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable.

- ● → Core point
- ● → Border point
- ● → Noise/Outliers

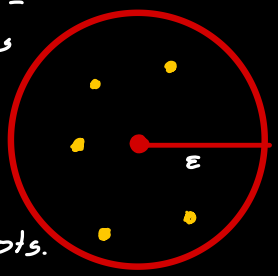
} Non-linear clustering.

Hyperparameters

minpts = 4 | ε = radius
(minimum points) (epsilon)

Core point

→ All minimum number of points one lies within circle. hence the given red point is said to be core point.



→ No. of points within the ε should be ≥ minpts. (including core point also)

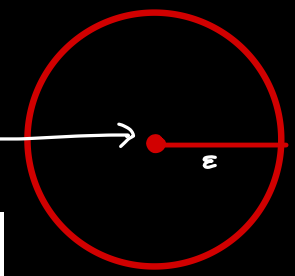
→ In above case, minpts = 6

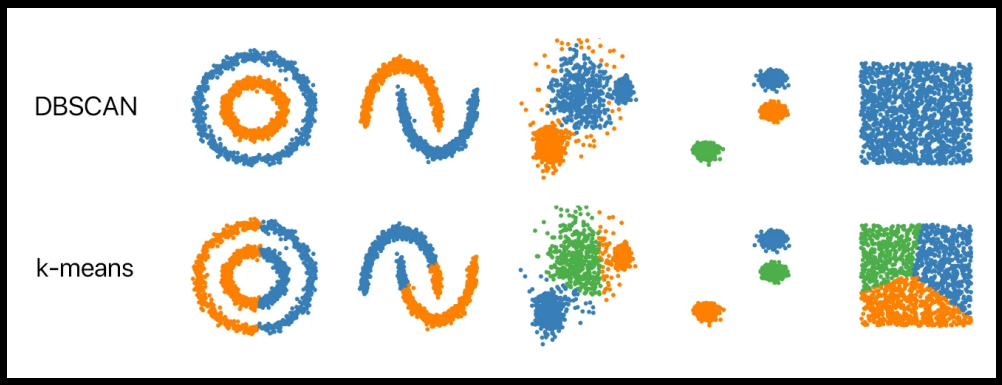
Border points

→ No. of data points within the radius will be < minpts = 4

Noise/Outliers

→ DBScan is robust for outliers.
→ No one points lies within radius(ε)





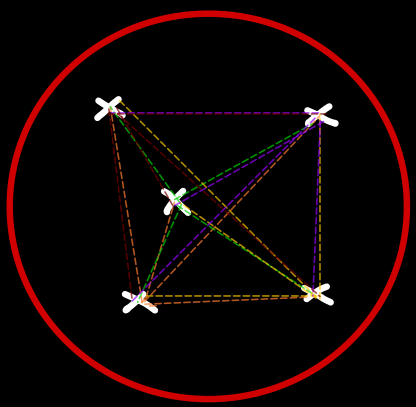
Silhouette Score

→ Clustering validation technique. (model validation technique)

For data point $i \in C_I$ (data point i in the cluster C_I), let

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j)$$

be the mean distance between i and all other data points in the same cluster, where $|C_I|$ is the number of points belonging to cluster C_I, and $d(i, j)$ is the distance between data points i and j in the cluster C_I (we divide by $|C_I| - 1$ because we do not include the distance $d(i, i)$ in the sum). We can interpret $a(i)$ as a measure of how well i is assigned to its cluster (the smaller the value, the better the assignment).

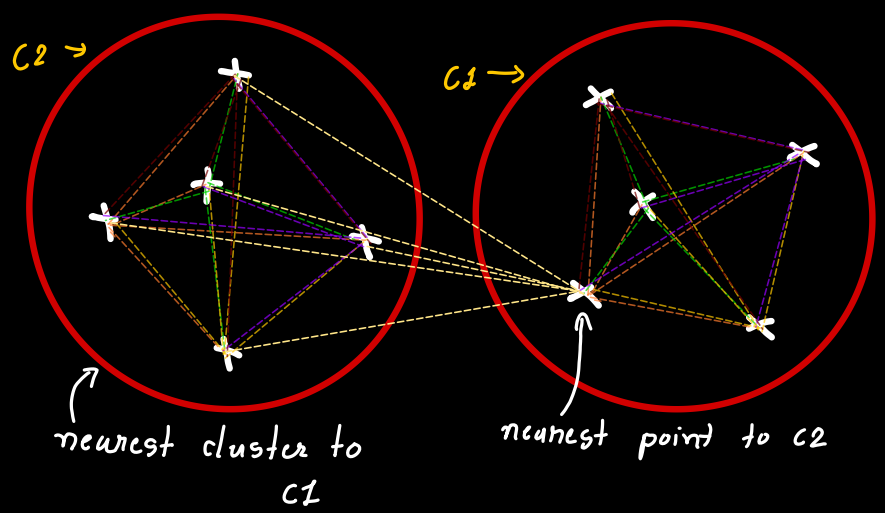


We then define the mean dissimilarity of point i to some cluster C_J as the mean of the distance from i to all points in C_J (where $C_J \neq C_I$).

For each data point $i \in C_I$, we now define

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j)$$

to be the *smallest* (hence the $\min$ operator in the formula) mean distance of i to all points in any other cluster (i.e., in any cluster of which i is not a member). The cluster with this smallest mean dissimilarity is said to be the "neighboring cluster" of i because it is the next best fit cluster for point i.



We now define a *silhouette* (value) of one data point i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_I| > 1$$

and

$$s(i) = 0, \text{ if } |C_I| = 1$$

Which can be also written as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

From the above definition it is clear that

$$-1 \leq s(i) \leq 1$$

if $a_i > b_i$ = 👌

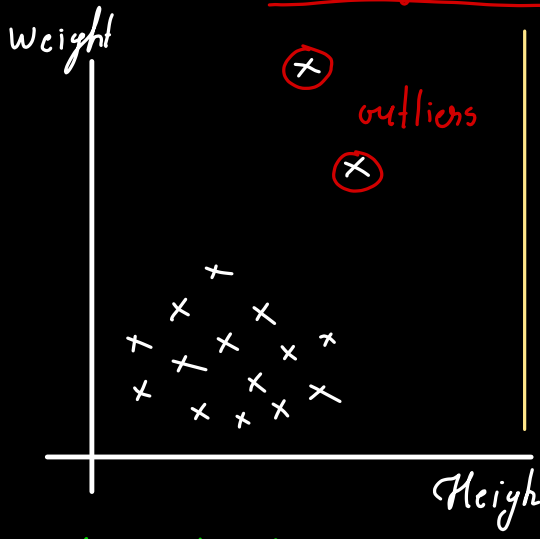
if $a_i < b_i$ = 👎

-1 to +1

≈ +1 = good

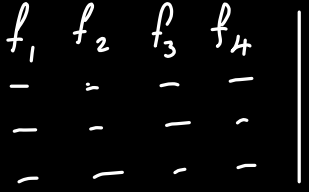
≈ -1 = bad

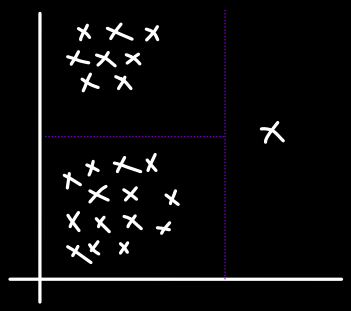
Anomaly Detection [To detect outliers]



1. Isolation forest (Decision tree)

→ our target is to find those data-point which are different from others.





→ Anomaly detection is the process of identifying rare events or outliers that deviate significantly from the norm in datasets. these anomalies can signal potential issues or interesting pattern in the data.

- **Applications**

Cybersecurity, User behaviour Analysis, Fraud detection, Healthcare

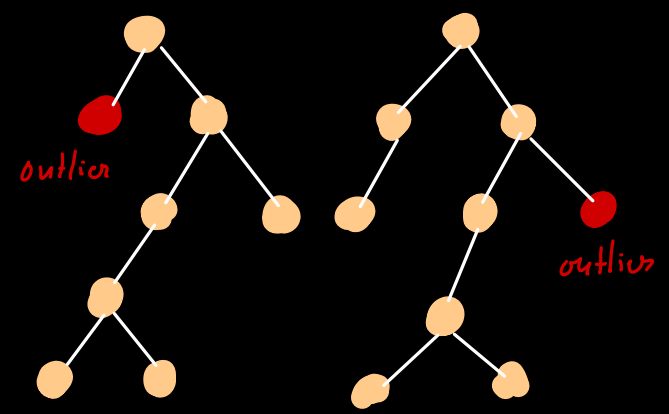
Anomaly detection technique

1. Isolation forest

→ Similar as random forest.

→ Internally, this method will create multipal isolation trees.

→ Anomaly score will help us to predict that whether a per-ticular node is outlier or not.



Anomaly score

$$S(x, m) = 2^{\frac{-E(h(x))}{c(m)}}$$

m : total n. of data points

x : Data point

$E(h(x))$: Average search depth for x from isolated tree.

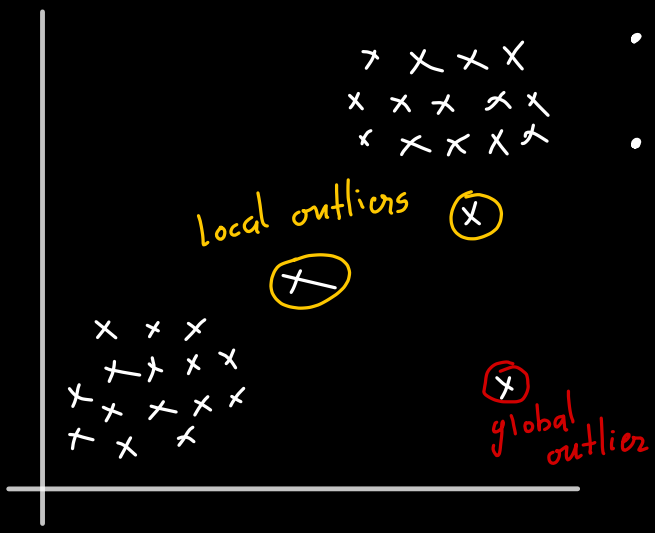
$c(m)$: Average depth of all data points

if $S(x,m) \approx 1$ = outlier

$S(x,m) \approx 0.5$ = normal data point

2. Using DBSCAN clustering

3. local outlier fector anomaly detection



- for global outliers we can use DBSCAN, Isolation forest.
- for local outliers, we can use local outlier fector.