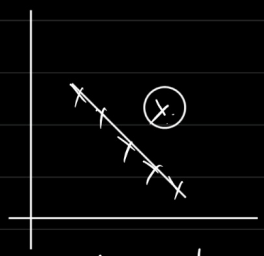
Anomaly detection (To detect outliers)

⇓

Abnormality

→ Anomaly detection algos are Unsupervised.



→ sometimes to detect outliers traditional methods like Box plot, violin plot fails.

Why?

→ In few case Anomaly plays an important role.

Ex1

transaction amount	Acc no
1000	—
2000	—
1500	—
3000	—
2 Lakhs	—

→ This customer being outlier, holds an important role. ⟹ Bank will target this customer for the loan.

∴ Outliers sometimes makes sense.

Ex2 → Fraud transaction.

tr_1 – Delhi
tr_2 – Delhi
tr_3 – Delhi
} Normal

$tran_4$ – USA ⟶ you receive a msg ⟹ anomaly.

Ex3 → When you login a gmail or any portal.

Ex4 VK. → 5, 10, 5, 4, 3, 5, 110 → This outlier but important.

Ex5 Cancer prediction ⇒ Normally people don't have Cancer.

Ex6 → Fraud IP.

* Detection of Outliers

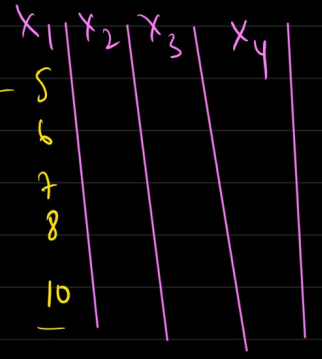
1. Isolation forest Anomaly detection
2. DBSCAN
3. Local Outlier Factor Anomaly detection.

1. Isolation forest (USL algorithms)

⇓

⇒ multiple Isolation trees

→ Isolation tree is like a decision tree

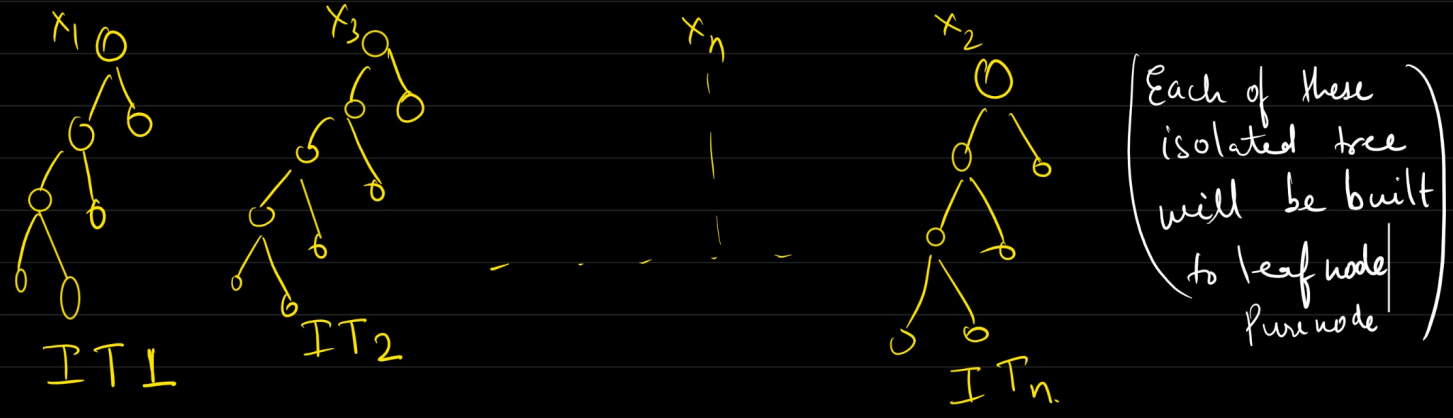


Construction of Isolation tree

→ Select a feature randomly X_1

→ Randomly choose a split value within the range of X_1

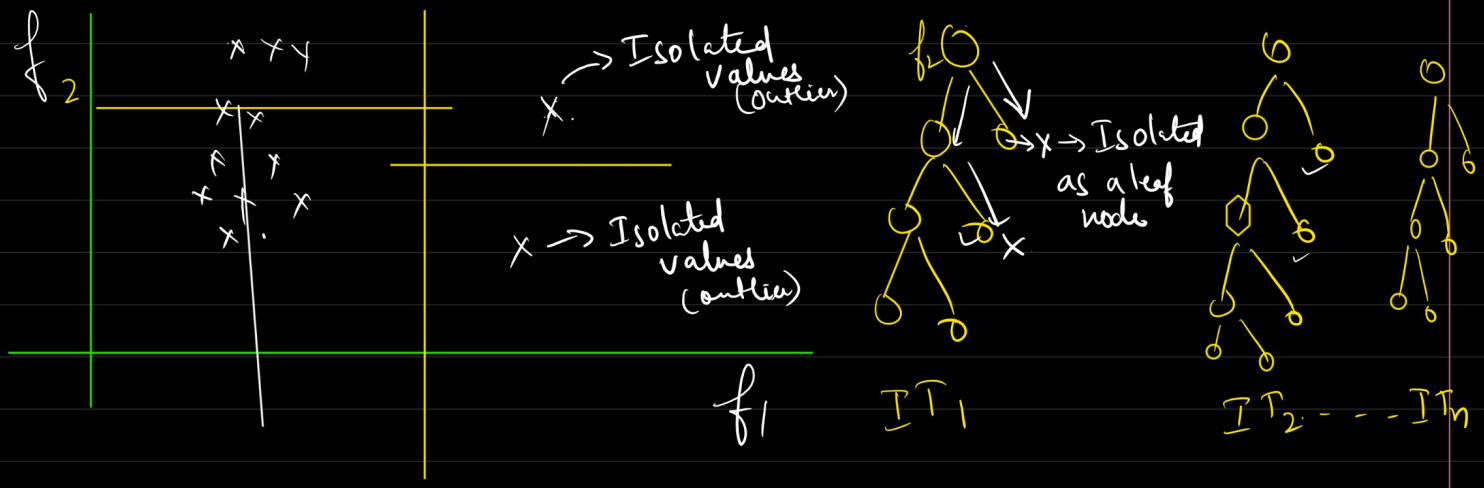
→ Repeat the process recursively to build the tree.



(Each of these isolated tree will be built to leaf node | Pure node)

* How you will get outlier?

→ Anomaly | outlier due to their distinctiveness, tend to end up in leaf nodes at shorte path.



* Since outliers are different from normal dp, during construction of isolated trees it will be isolated as a seperate leaf node.

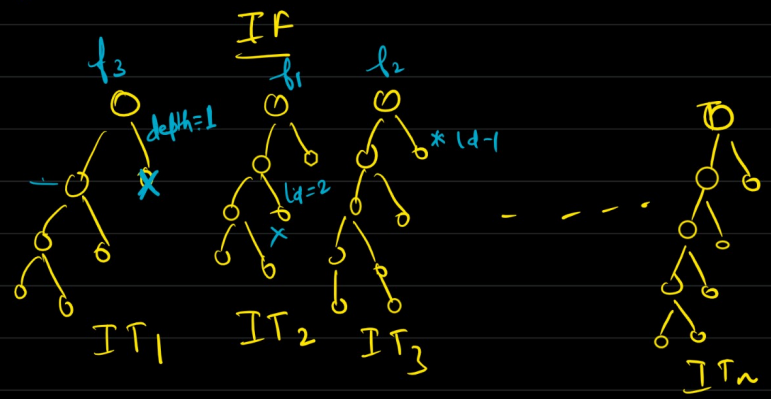
* **Anomaly score**

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

where m is total no of datapoints
x → datapoint for which you want to check anomaly score.

$E(h(x))$ = Average search depth of x in all of the isolated tree.
↓ for one dp

$C(m)$ = Average depth of all the dp's in all Isolation tree



$E(h(x)) \ll C(m)$ Since C(m) is avg depth for all the dp's so C(m) will be far greater than h(x)

$S(x,m) := 2^{\frac{-E(h(x))}{C(m)}}$ → will be very small value

$S(x,m) \approx 1 \Rightarrow$ Anomaly score $\Rightarrow$ outlier

Generally If $S(x,m) \geq 0.5 \Rightarrow$ outliers.

threshold

$E(h(x)) >> c(m) \Rightarrow S(x,m) \approx 0.5 \Rightarrow$ Normal dp's

< 0.5

Isolation Forest is an anomaly detection algorithm that isolates outliers by randomly partitioning data. It constructs multiple isolation trees by recursively splitting data using random features and split values. Anomalies, being distinct and few, tend to be isolated with fewer splits, leading to shorter paths in these trees. The algorithm calculates an anomaly score based on the average path length, where a higher score indicates a higher likelihood of the point being an anomaly. Isolation Forest is efficient, does not assume any data distribution, and performs well with high-dimensional data, making it ideal for large-scale anomaly detection tasks.

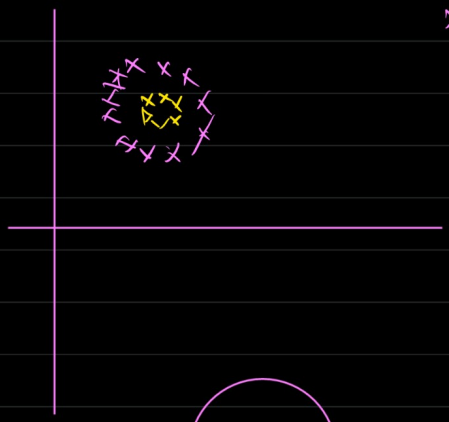
DBSCAN (Density-Based Spatial Clustering of Applications with Noise) can be used for anomaly detection by identifying points that do not belong to any dense cluster. It groups data points based on their density, using parameters eps (neighborhood radius) and minPts (minimum points in a neighborhood). Points that are isolated and do not meet these criteria are classified as noise, which are considered anomalies. DBSCAN is effective because it doesn't require specifying the number of anomalies, can detect outliers in clusters of arbitrary shapes, and is robust to noise, making it suitable for complex, real-world datasets.

Local Outlier Factor (LOF) is an anomaly detection algorithm that identifies outliers by comparing the local density of a data point with that of its neighbors. It calculates a LOF score, where a score significantly greater than 1 indicates the point is an outlier. The algorithm considers the distance to a point's k-nearest neighbors to determine its local reachability density. LOF is effective in datasets with varying density, as it detects outliers based on the local context rather than global distribution. It's commonly used in fraud detection, network security, and other applications requiring the identification of subtle anomalies.

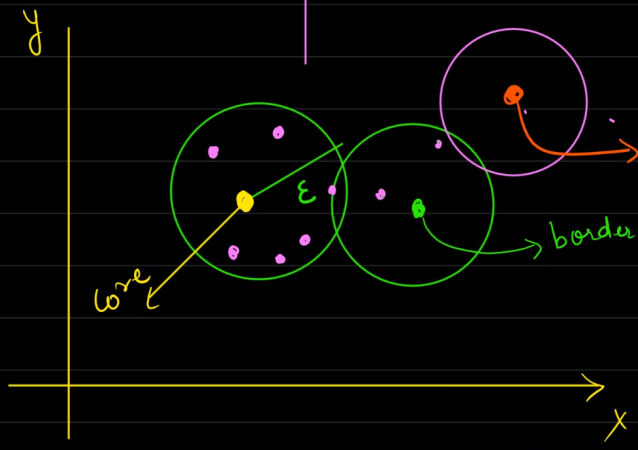
DBSCAN (density based Spatial Clustering of Application with noise)

* Characteristics of DBSCAN
  - → Finds groups / pattern
  - → Finds outliers / noise

if already we had Kmean / hierarchical, then why DBSCAN?



* Distance based Algorithm such as K means & hierarchical can not be used here (non linear data / non linear clustering)



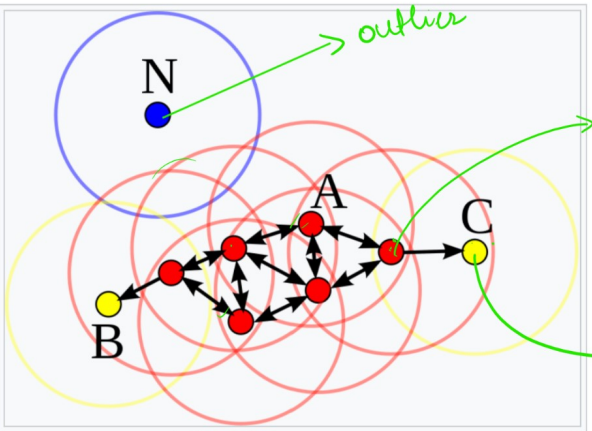
- ● → Core data point
- ● → border data point
- ● → outlier dp.

* minimum no of dp = 4
* radius (ε) = 1 m

* Core dp :-
  The dp's which hv atleast minimum no. of dp in its radius ε (4)

* Border dp → no. of dp within its boundary (ε) will be less than minimum dp

* Outlier (Noise) → No dp in radius ε



outlier

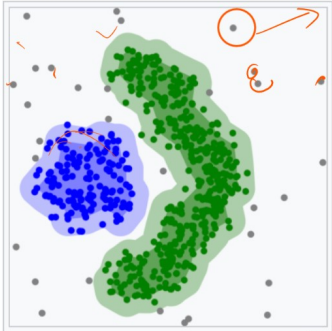
core dp

border dp

In this diagram, $\text{minPts} = 4$. 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.

* Some Examples of DBSCAN working very well with non linear data.

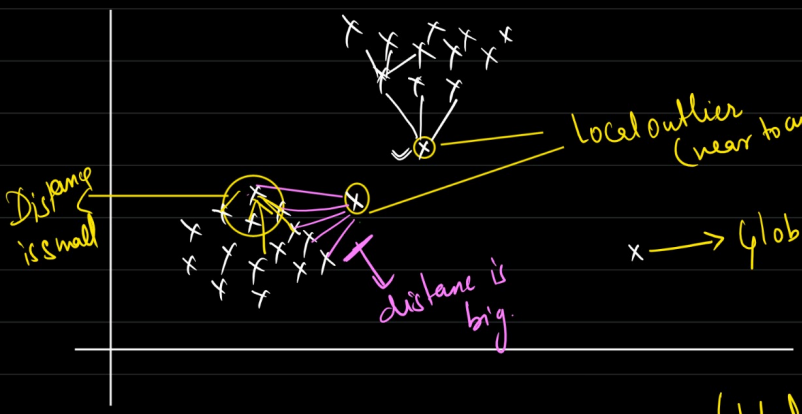
outlier



DBSCAN can find non-linearly separable clusters. This dataset cannot be adequately clustered with k-means or Gaussian Mixture EM clustering.

The left image depicts a more traditional clustering method that does not account for multi-dimensionality. Whereas the right image shows how DBSCAN can contort the data into different shapes and dimensions in order to find similar clusters.

Local Outlier factor Anomaly detection



① Local outlier
② Global outlier

→ Global outlier is easy to detect using Isolation forest / DBSCAN.

→ How to detect local outlier ?

⇒ Using Local outlier factor.

|| local density. Using KNN.

* Calculate distance of a dp with its k- nearest neighbour.
* dp's which are far will have more distance from its neighbours. and vice - versa

* Higher distance → density less ⇒ Outlier

LOF score ⇒ $\frac{\text{dp of Interest distance}}{\text{All dp's distance}}$

⇓

> threshold ⇒ outlier.