Isolation Forest - *(Anomalies Detection)*

It is an unsupervised machine learning algorithm for anomaly detection that works by isolating outliers instead of modelling normal data.

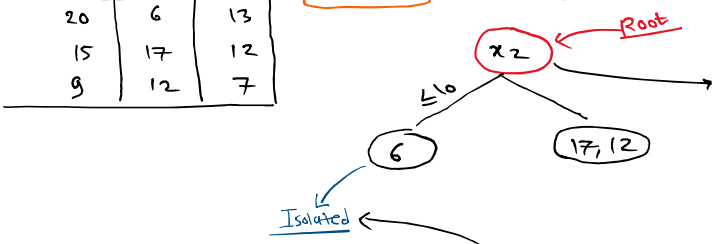
How it works

- **Random partitioning**: The algorithm recursively partitions the data by randomly selecting a feature and a random split point within the range of that feature.
- **Tree creation**: This random splitting process is repeated to build an "isolation tree." Multiple trees are created to form the "isolation forest".
- **Isolation of anomalies**: Because anomalies are few and different, they tend to be separated from the rest of the data in fewer splits than normal data points.
- **Anomaly score calculation**: An anomaly score is assigned to each data point based on its average path length across all trees in the forest. Data points with shorter average path lengths are considered anomalies.
- **Efficiency**: This method is efficient for large datasets because it does not need to build a model of the normal data points, and it uses random splits instead of density estimation

Isolation Forest (unsupervised algorithm / no target column required)

x_1	x_2	x_3
20	6	13
15	17	12
9	12	7

STEP-1: select any feature randomly (i.e. let's say: x_2) for root node.



$\in [\text{Min } x_2 = 6 \mid \text{Max } x_2 = 17]$

Pick criteria in betⁿ. min & max, e.g. 10 (i.e. $x_2 \leq 10$)

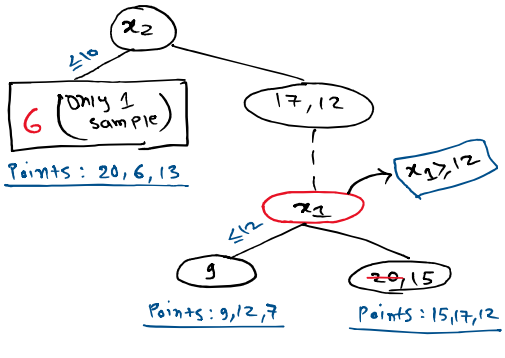
* **STEP-2**: Now we used 1st row (20, 6, 13) for x_2 split

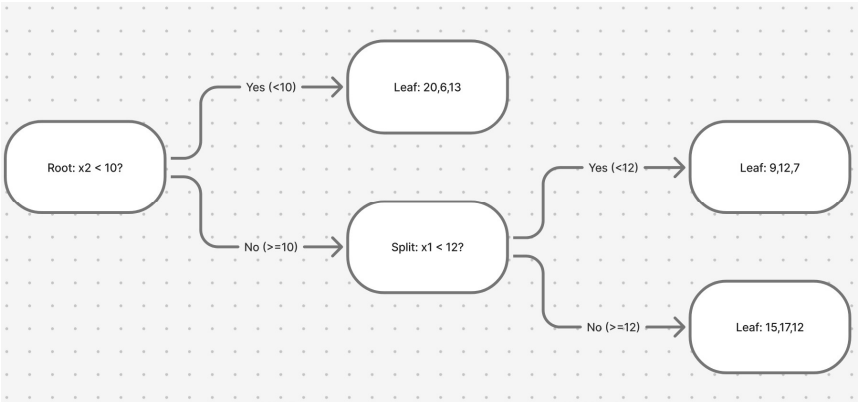
let's choose another root randomly i.e. x_1 & this time we not considering row 1. because it was used before, so that for x_1 20 is not used.

$\in [\text{Min}(x_1) = 9 \mid \text{Max}(x_1) = 15]$ | $(x_1 = \cancel{20}, 15, 9)$

now choose splitting criterian randomly in betⁿ. range of Min(x_1) & Max(x_1) = 12

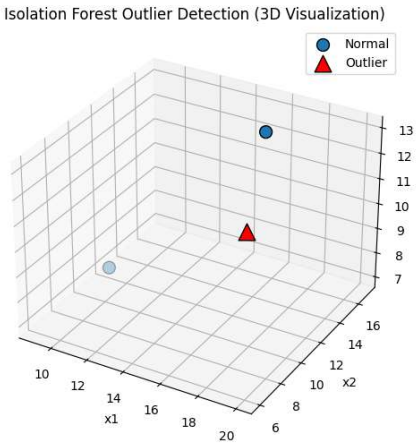
Now, our tree looks like:





* **Conclusion**:
- like in row 1st {20, 6, 13} → Anomalies can detect easily, isolate quickly.
- Normal point needs more splits.

Points	Depth	Interpretation
(20, 6, 13)	1 (Fast splitted)	Very easy to isolate (→ likely outlier)
(9, 12, 7)	2	Less isolated
(15, 17, 12)	2	Less isolated




```
scores = isolation_forest.decision_function(df)
scores
```

```
array([-0.00212643,  0.18278396,  0.14418879])
```

```
pred = isolation_forest.predict(df)
pred
```

```
array([-1,  1,  1])
```

All are:

- positive
- close to zero
- close to each other

In Isolation Forest:

- Large negative score → strongly anomalous
- Small positive score → normal but near boundary
- Large positive → very normal