Decision Tree

Decision Tree

A Decision Tree is a supervised Machine learning algorithm.

It is used in both classification and regression algorithms.

The decision tree is like a tree with nodes.

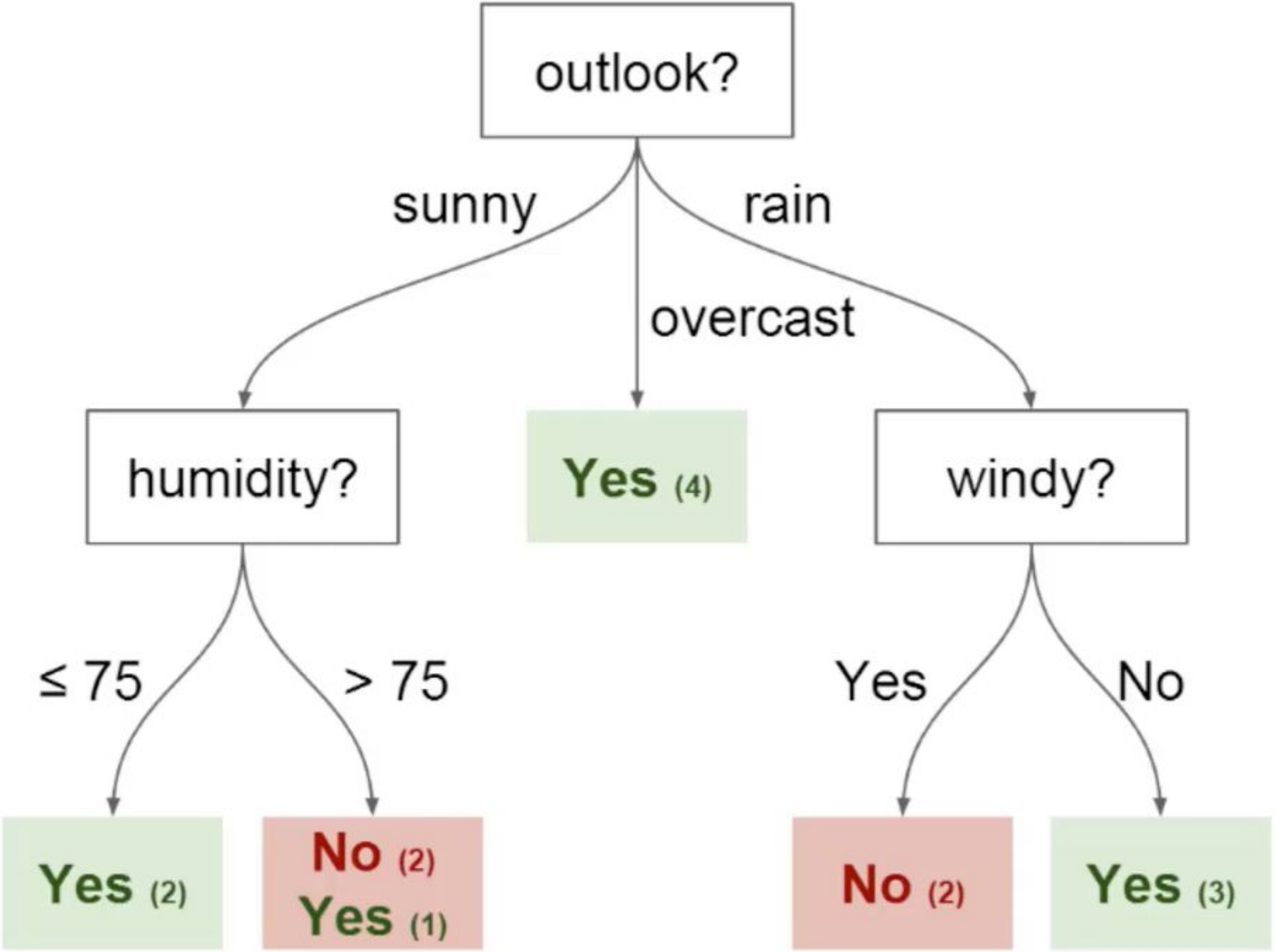
What is a Decision Tree?

Imagine that you play football every Sunday and you always invite your friend to come to play with you.

Sometimes your friend actually comes and sometimes he doesn't. The factor on whether or not to come depends on numerous things, like weather, temperature, wind, and fatigue.

We start to take all of these features into consideration and begin tracking them alongside your friend's decision whether to come for playing or not.

You can use this data to predict whether or not your friend will come to play football or not.



outlook?
sunny
overcast
rain
humidity?
Yes (4)
windy?
≤ 75
> 75
Yes
No
Yes (2)
No (2)
Yes (1)
No (2)
Yes (3)

Decision Tree

Every decision tree consists following list of elements:

- Node
- Edges
- Root
- Leaves

Building Decision Tree from Scratch

While building a Decision tree, the main thing is to select the best attribute from the total features list of the dataset for the root node as well as for sub-nodes.

The selection of best attributes is being achieved with the help of a technique known as the Attribute selection measure (ASM).

With the help of ASM, we can easily select the best features for the respective nodes of the decision tree.

There are two techniques for ASM:

a) Information Gain

b) Gini Index

Entropy

Entropy is a degree of randomness or uncertainty or disorder.

Entropy = 0 means no impurity.

Higher the entropy means higher the impurity.

Lower entropy means more pure node and higher entropy means less pure nodes.

$$H(X) = H(p_1, \ldots, p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Entropy

Calculate the entropy for following sets

a. X = {a,a,a,a,b,b,b,b}
b. X = {a,a,a,a}
c. X = {a,a,b,b,c,c,d,d}

Entropy

Calculate the entropy for following sets

a. $X = \{a,a,a,a,b,b,b,b\} \rightarrow 1$
b. $X = \{a,a,a,a\} \rightarrow 0$
c. $X = \{a,a,b,b,c,c,d,d\} \rightarrow 2$

Information Gain

IG calculates effective change in entropy after making a decision based on the value of an attribute.

For decision trees, it’s ideal to base decisions on the attribute that provides the largest change in entropy, the attribute with the highest gain.

IG is used as ASM in ID3(Iterative Dichotomiser 3) algorithm.

Information Gain

$$IG(T,A) = Entropy(T) - \sum_{v \in A} \frac{|T_v|}{T} \cdot Entropy(T_v)$$

T = Target

A = column/feature

v = each value in A

Dataset

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

ID3 Algorithm

$H(S) = - p(yes) * \log_2(p(yes)) - p(no) * \log_2(p(no))$

$= - (9/14) * \log_2(9/14) - (5/14) * \log_2(5/14)$

$= - (-0.41) - (-0.53) = 0.94$

ID3 Algorithm

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Outlook

Categorical values - sunny, overcast and rain

H(Outlook = sunny) = -(2/5)*log(2/5)-(3/5)*log(3/5) =0.971

H(Outlook = rain) = -(3/5)*log(3/5)-(2/5)*log(2/5) =0.971

H(Outlook = overcast) = -(4/4)*log(4/4)-0 = 0

ID3 Algorithm

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Outlook

Categorical values - sunny, overcast and rain

H(Outlook = sunny) = -(2/5)*log(2/5)-(3/5)*log(3/5) =0.971

H(Outlook = rain) = -(3/5)*log(3/5)-(2/5)*log(2/5) =0.971

H(Outlook = overcast) = -(4/4)*log(4/4)-0 = 0

Information Gain = H(S) - p(sunny) * H(Outlook=sunny) - p(rain) * H(Outlook=rain) - p(overcast) * H(Outlook=overcast)

= 0.94 - (5/14)*0.971 - (5/14)*0.971 - (4/14)*0

=0.94-0.693

= 0.247

ID3 Algorithm

Temperature

Categorical values - hot, mild, cool

H(Temperature = hot) = -(2/4)*log(2/4)-(2/4)*log(2/4) = 1

H(Temperature = cool) = -(3/4)*log(3/4)-(1/4)*log(1/4) = 0.811

H(Temperature = mild) = -(4/6)*log(4/6)-(2/6)*log(2/6) = 0.9179

Information Gain = H(S) - p(hot)*H(Temperature=hot) - p(mild)*H(Temperature=mild) - p(cool)*H(Temperature=cool)

= 0.94 - (4/14)*1 - (6/14)*0.9179 - (4/14)*0.811 = 0.0292

ID3 Algorithm

Humidity

Categorical values - high, normal

H(Humidity=high) = -(3/7)*log(3/7)-(4/7)*log(4/7) = 0.983

H(Humidity=normal) = -(6/7)*log(6/7)-(1/7)*log(1/7) = 0.591

Information Gain = H(S) - (high)*H(Humidity=high) - p(normal)*H(Humidity=normal)

=0.94 - (7/14)*0.983 - (7/14)*0.591

= 0.94 - 0.787

= 0.153

ID3 Algorithm

Wind

Categorical values - weak, strong

H(Wind=weak) = -(6/8)*log(6/8)-(2/8)*log(2/8) = 0.811

H(Wind=strong) = -(3/6)*log(3/6)-(3/6)*log(3/6) = 1

Information Gain = H(S) - p(weak)*H(Wind=weak) - p(strong)*H(Wind=strong) =

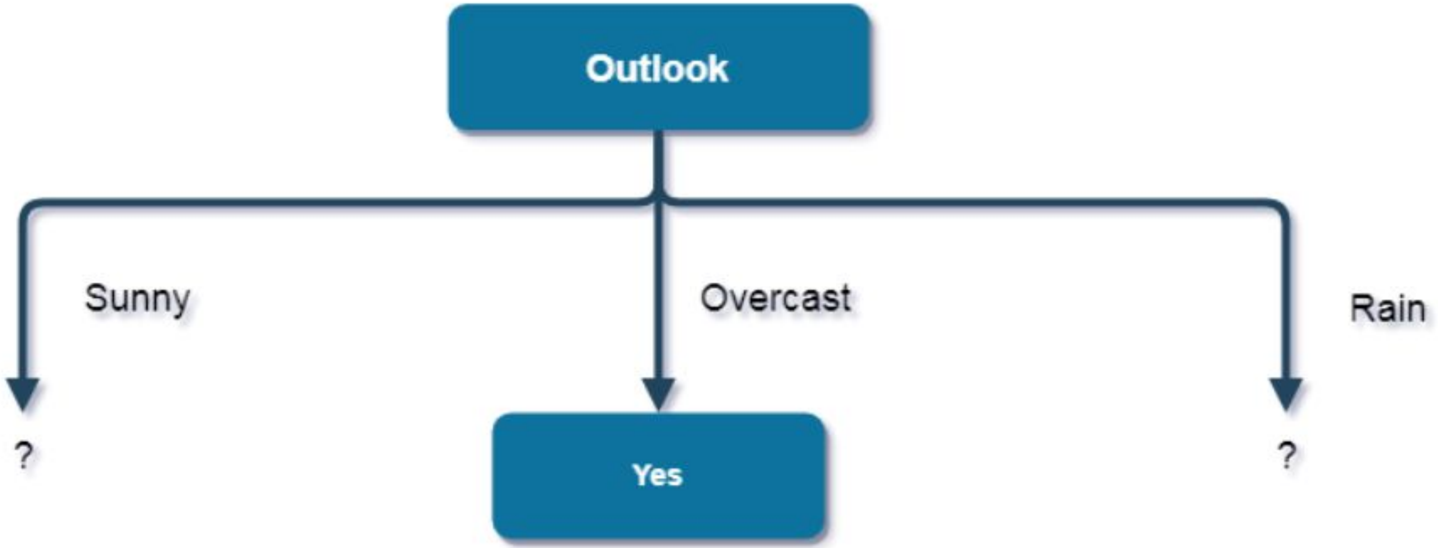
= 0.94 - (8/14)*0.811 - (6/14)*1

0.94 - 0.892 = 0.048

ID3 Algorithm

Since Outlook has maximum IG we use Outlook.

So decision tree built so far is:



Gini Index/Coefficient

Gini is a measure of impurity/ purity used while creating a decision tree in the CART(known as Classification and Regression Tree) algorithm.

An attribute having a low Gini index value should be preferred in contrast to the high Gini index value.

It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.

Gini Index

$$GINI(t) = 1 - \sum_{k} \left[p(C_k | t)^2 \right]$$

- $p(Ck|t)$ is the probability of a node t being the category Ck.
- The Gini coefficient of the node t is 1 minus the sum of the probability of all the categories.

Gini Index

Training Data				
Date	Weather	Temperature	Wind Level	Go out for running?
2020-01-01	Sunny	High	Low	No
2020-01-02	Sunny	Medium	Medium	Yes
2020-01-03	Cloudy	High	Medium	Yes
2020-01-04	Cloudy	Medium	High	Yes
2020-01-05	Rainy	High	Low	No
2020-01-06	Rainy	High	Medium	No
2020-01-07	Sunny	Low	High	No

Gini Index

Training Data	
Weather	Go out for running?
Sunny	No
Sunny	Yes
Cloudy	Yes
Cloudy	Yes
Rainy	No
Rainy	No
Sunny	No

Totally 3 samples,

1 "Yes" and 2 "No"

Training Data	
Weather	Go out for running?
Sunny	No
Sunny	Yes
Cloudy	Yes
Cloudy	Yes
Rainy	No
Rainy	No
Sunny	No

Totally 2 samples,

2 "Yes" and 0 "No"

Training Data	
Weather	Go out for running?
Sunny	No
Sunny	Yes
Cloudy	Yes
Cloudy	Yes
Rainy	No
Rainy	No
Sunny	No

Totally 2 samples,

0 "Yes" and 2 "No"

Gini Index

$$GINI(weather = sunny) = 1 - \left[\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right] = 0.444$$

$$GINI(weather = cloudy) = 1 - \left[\left(\frac{2}{2}\right)^2 + \left(\frac{0}{2}\right)^2\right] = 0$$

$$GINI(weather = rainy) = 1 - \left[\left(\frac{0}{2}\right)^2 + \left(\frac{2}{2}\right)^2\right] = 0$$

P(Yes) P(No)

Gini Index

$$GINI(T) = \frac{|t_{\text{left}}|}{|T|} \times Gini(t_{\text{left}}) + \frac{|t_{\text{right}}|}{|T|} \times Gini(t_{\text{right}})$$

- |t_left| and |t_right| are the sample size of the nodes on the left and right respectively.
- |T| is the sample size of the candidate parent node T.

Gini Index

1. Split by “sunny” and “not sunny”, Gini coefficient = 0.476

Training Data

Weather	Go out for running?
Sunny	No
Sunny	Yes
Cloudy	Yes
Cloudy	Yes
Rainy	No
Rainy	No
Sunny	No

$|t_{\text{left}}| = 3$

$|t_{\text{right}}| = 4$

$$GINI(weather_1) = \frac{|t_{\text{left}}|}{|T|} \times Gini(weather = sunny) + \frac{|t_{\text{right}}|}{|T|} \times Gini(t_{\text{right}})$$

$$GINI(weather_1) = \frac{3}{7} \times 0.444 + \frac{4}{7} \times \left\{1 - \left[\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right]\right\}$$

$$GINI(weather_1) = 0.476$$

2 "yes" and 2 "no" left

Gini Index

2. Split by "cloudy" and "not cloudy", Gini coefficient = 0.229

Training Data

Weather	Go out for running?
Sunny	No
Sunny	Yes
Cloudy	Yes
Cloudy	Yes
Rainy	No
Rainy	No
Sunny	No

$|t_{\text{left}}| = 2$

$|t_{\text{right}}| = 5$

$$GINI(weather_2) = \frac{|t_{\text{left}}|}{|T|} \times Gini(weather = cloudy) + \frac{|t_{\text{right}}|}{|T|} \times Gini(t_{\text{right}})$$

$$GINI(weather_2) = \frac{2}{7} \times 0 + \frac{5}{7} \times \left\{1 - \left[\left(\frac{1}{5}\right)^2 + \left(\frac{4}{5}\right)^2\right]\right\}$$

$$GINI(weather_2) = 0.229$$

1 "yes" and 4 "no" left

Gini Index

3. Split by “rainy” and “not rainy”, Gini coefficient = 0.343

Training Data	
Weather	**Go out for running?**
Sunny	No
Sunny	Yes
Cloudy	Yes
Cloudy	Yes
Rainy	No
Rainy	No
Sunny	No

$$|t_{\text{left}}| = 2$$

$$|t_{\text{right}}| = 5$$

$$GINI(weather_3) = \frac{|t_{\text{left}}|}{|T|} \times Gini(weather = rainy) + \frac{|t_{\text{right}}|}{|T|} \times Gini(t_{\text{right}})$$

$$GINI(weather_3) = \frac{2}{7} \times 0 + \frac{5}{7} \times \left\{1 - \left[\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right]\right\}$$

$$GINI(weather_3) = 0.343$$

3 "yes" and 2 "no" left

Gini Index

The same logic will be applied to all the other possibilities until we found the minimum Gini coefficient, and that will be the root node. Similarly, for the internal nodes, the same logic is applied to make sure every split have least Gini coefficient.

How Does the Decision Tree Algorithm works?

The basic idea behind any decision tree algorithm is as follows:

1. Select the best Feature using Attribute Selection Measures(ASM) to split the records.

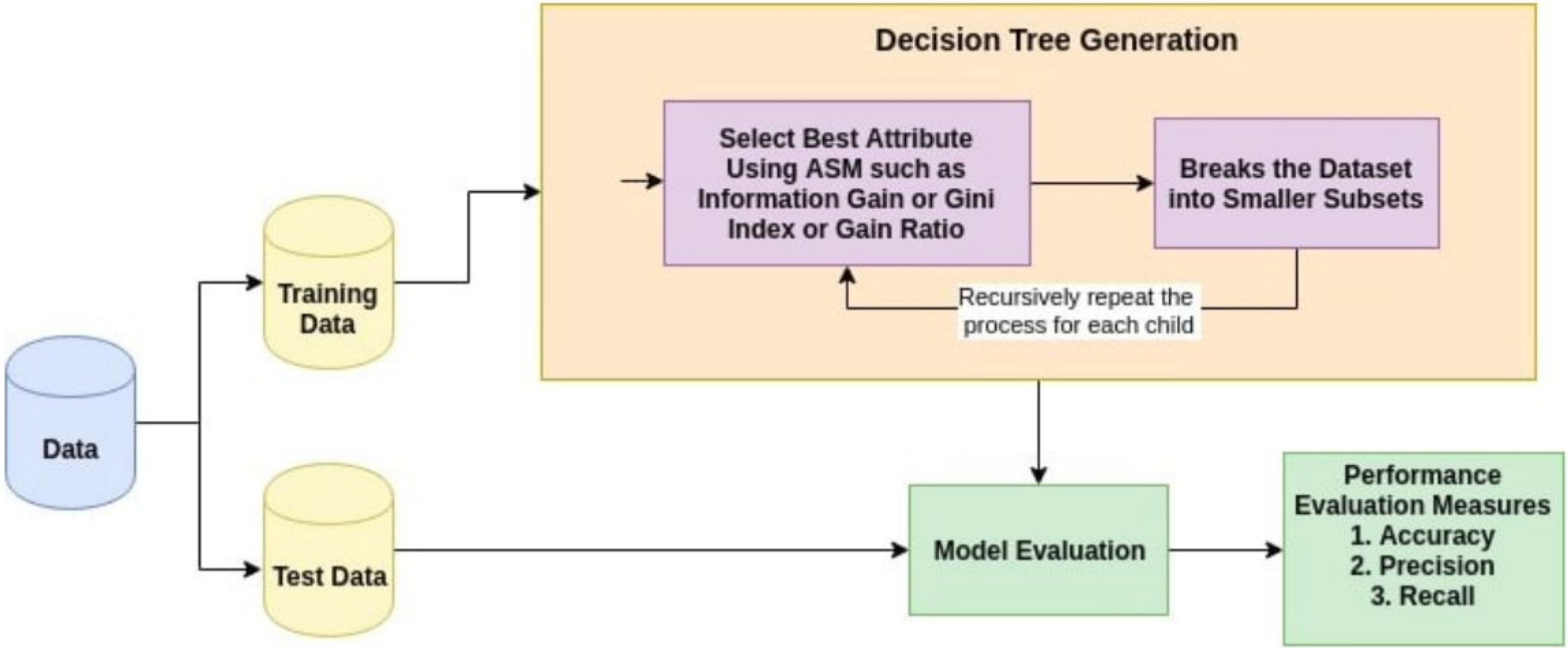
2. Make that attribute/feature a decision node and break the dataset into smaller subsets.

3. Start the tree-building process by repeating this process recursively for each child until one of the following condition is being achieved :

   a) All tuples belonging to the same attribute value.

   b) There are no more of the attributes remaining.

How Does the Decision Tree Algorithm works?



References

https://www.analyticsvidhya.com/blog/2021/04/beginners-guide-to-decision-tree-classification-using-python/

https://iq.opengenus.org/id3-algorithm/

https://www.youtube.com/watch?v=g9c66TUylZ4

https://towardsdatascience.com/understanding-random-forest-58381e0602d2

https://www.analyticsvidhya.com/blog/2020/10/all-about-decision-tree-from-scratch-with-python-implementation/

https://stats.stackexchange.com/questions/91044/how-to-calculate-precision-and-recall-in-a-3-x-3-confusion-matrix