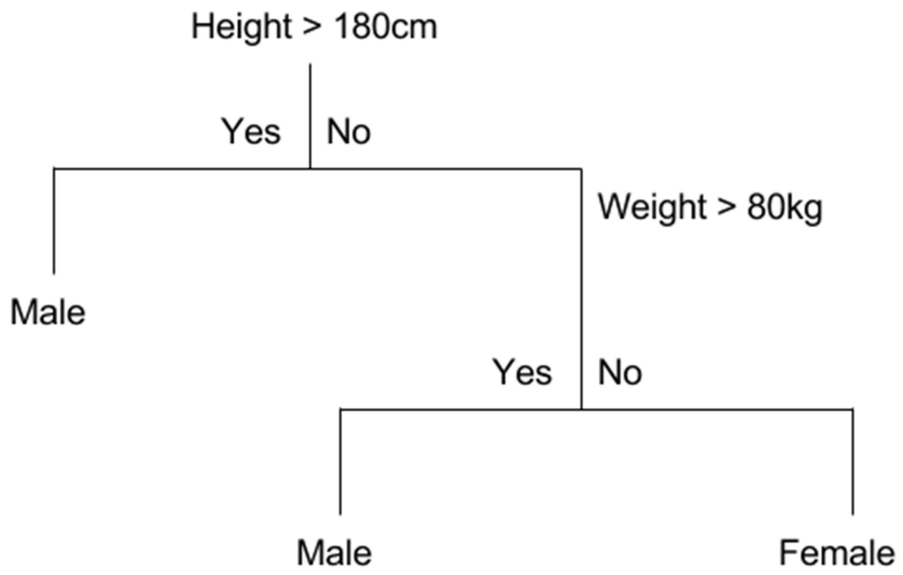
What is a decision tree?

A decision tree is a classification and prediction tool having a tree-like structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.



Above we have a small decision tree. An important advantage of the decision tree is that it is highly interpretable. Here If Height > 180cm or if height < 180cm and weight > 80kg person is

male. Otherwise female. Did you ever think about how we came up with this decision tree? I will try to explain it using the weather dataset.

Before going to it further I will explain some important terms related to decision trees.

Entropy

In machine learning, entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information.

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$$

Information Gain

Information gain can be defined as the amount of information gained about a random variable or signal from observing another random variable. It can be considered as the difference between the entropy of parent node and weighted average entropy of child nodes.

$$IG(S, A) = H(S) - H(S, A)$$

Alternatively,

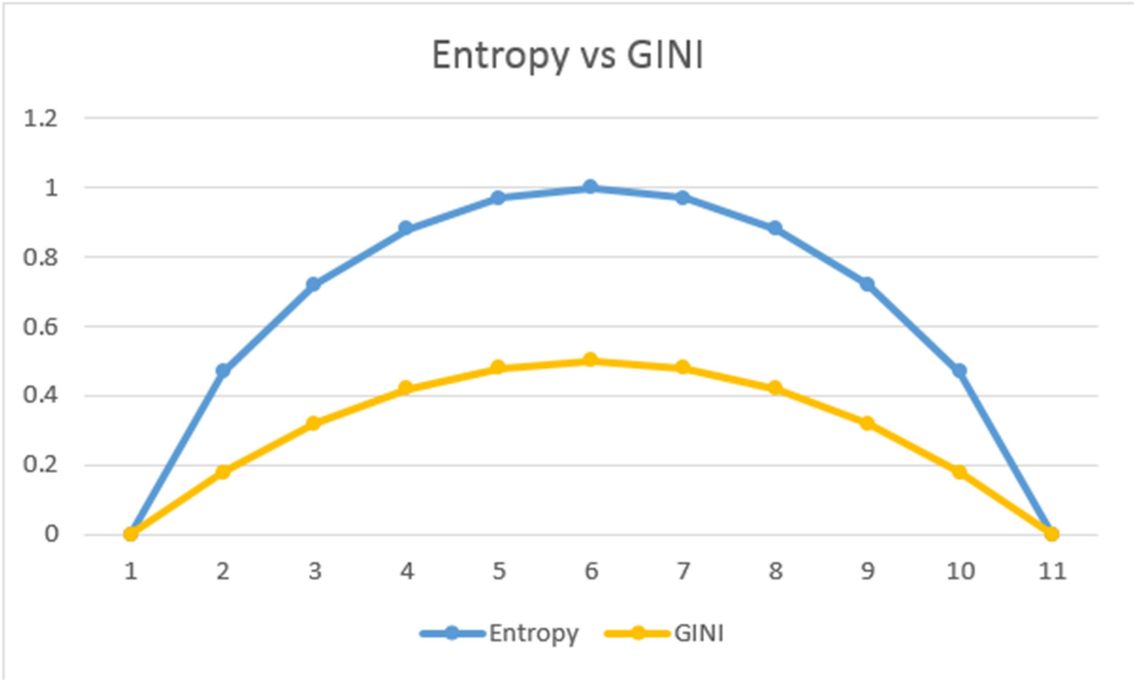
$$IG(S, A) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$

Gini Impurity

Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.

$$Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$$

Gini impurity is lower bounded by 0, with 0 occurring if the data set contains only one class.



There are many algorithms there to build a decision tree. They are

1. **CART** (Classification and Regression Trees) — This makes use of Gini impurity as the metric.
2. **ID3** (Iterative Dichotomiser 3) — This uses entropy and information gain as metric.

In this article, I will go through ID3. Once you got it, it is easy to implement the same using CART.

Classification using the ID3 algorithm

Consider whether a dataset based on which we will determine whether to play football or not.

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Here There are for independent variables to determine the dependent variable. The independent variables are Outlook, Temperature, Humidity, and Wind. The dependent variable is whether to play football or not.

As the first step, we have to find the parent node for our decision tree. For that follow the steps:

Find the entropy of the class variable.

E(S) = -[(9/14)log(9/14) + (5/14)log(5/14)] = 0.94

Note: Here typically we will take log to base 2.Here total there are 14 yes/no. Out of which 9 yes and 5 no. Based on it we calculated probability above.

From the above data for outlook we can arrive at the following table easily.

		play		
		yes	no	total
	sunny	3	2	5
Outlook	overcast	4	0	4
	rainy	2	3	5
				14

Now we have to calculate average weighted entropy, ie, we have found the total of weights of each feature multiplied by probabilities.

E(S, outlook) = (5/14)*E(3,2) + (4/14)*E(4,0) + (5/14)*E(2,3) = (5/14)(-(3/5)log(3/5)-(2/5)log(2/5))+ (4/14)(0) + (5/14)((2/5)log(2/5)-(3/5)log(3/5)) = 0.693

The next step is to find the information gain.

It is the difference between parent entropy and average weighted entropy we found above.

IG(S, outlook) = 0.94 - 0.693 = 0.247

Similarly find Information gain for Temperature, Humidity, and Windy.

IG(S, Temperature) = 0.940 - 0.911 = 0.029

IG(S, Humidity) = 0.940 - 0.788 = 0.152

IG(S, Windy) = 0.940 - 0.8932 = 0.048

Now select the feature having the largest entropy gain.

Here it is Outlook. So it forms the first node (root node) of our decision tree.

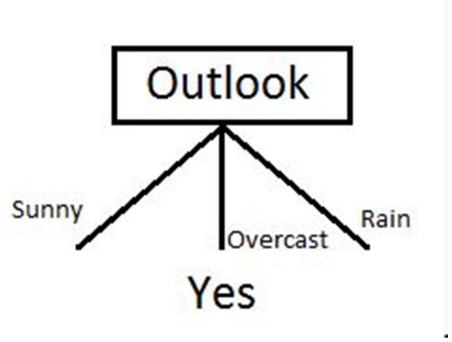
Now our data look as follows:

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Overcast	Hot	High	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Rain	Mild	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Since overcast contains only examples of class 'Yes' we can set it as yes. That means If outlook is overcast football will be played. Now our decision tree looks as follows.



The next step is to find the next node in our decision tree. Now we will find one under sunny. We have to determine which of the following Temperature, Humidity or Wind has higher information gain.

Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

Calculate parent entropy E(sunny)

E(sunny) = (-(3/5)log(3/5)-(2/5)log(2/5)) = 0.971.

Now Calculate the information gain of Temperature. IG(sunny, Temperature)

		play		
		yes	no	total
	hot	0	2	2
Temperature	cool	1	1	2
	mild	1	0	1
				5

E(sunny, Temperature) = (2/5)*E(0,2) + (2/5)*E(1,1) + (1/5)*E(1,0)=2/5=0.4

Now calculate information gain.

IG(sunny, Temperature) = 0.971–0.4 =0.571

Similarly, we get:

IG(sunny, Humidity) = 0.971

IG(sunny, Windy) = 0.020

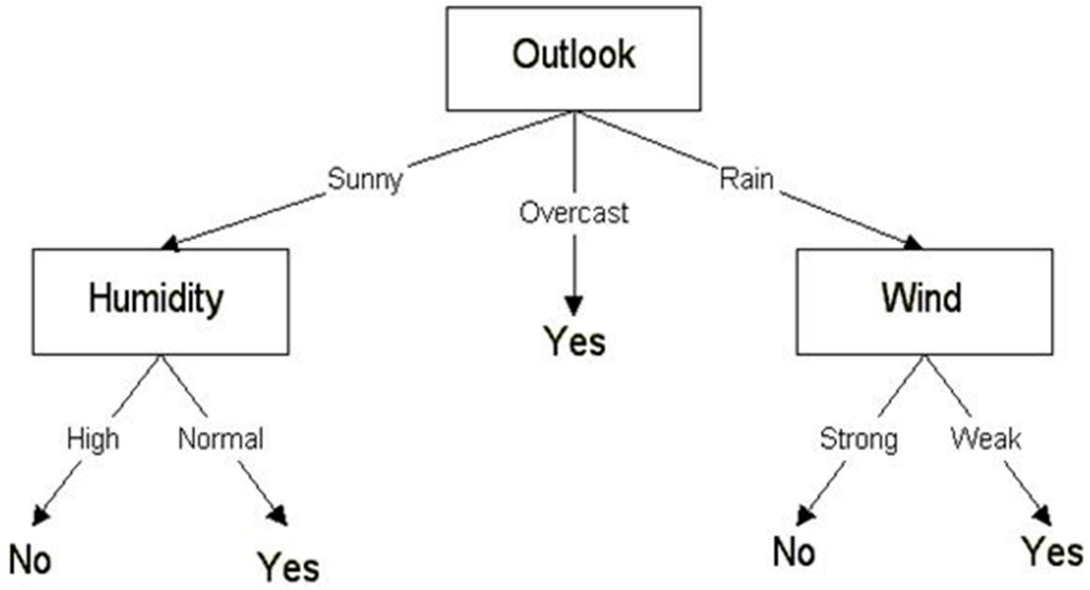
Here IG(sunny, Humidity) is the largest value. So Humidity is the node that comes under sunny.

	play	
Humidity	yes	no
high	0	3
normal	2	0

For humidity from the above table, we can say that play will occur if humidity is normal and will not occur if it is high. Similarly, find the nodes under rainy.

Note: A branch with entropy more than 0 needs further splitting.

Finally, our decision tree will look as below:



Classification using CART algorithm

Classification using CART is similar to it. But instead of entropy, we use Gini impurity.

So as the first step we will find the root node of our decision tree. For that Calculate the Gini index of the class variable

$Gini(S) = 1 - [(9/14)^2 + (5/14)^2] = 0.4591$

As the next step, we will calculate the Gini gain. For that first, we will find the average weighted Gini impurity of Outlook, Temperature, Humidity, and Windy.

First, consider case of Outlook:

		play		
		yes	no	total
	sunny	3	2	5
Outlook	overcast	4	0	4
	rainy	2	3	5
				14

$Gini(S, outlook) = (5/14)gini(3,2) + (4/14)*gini(4,0) + (5/14)*gini(2,3) = (5/14)(1 - (3/5)^2 - (2/5)^2) + (4/14)*0 + (5/14)(1 - (2/5)^2 - (3/5)^2) = 0.171+0+0.171 = 0.342$

$Gini\ gain\ (S, outlook) = 0.459 - 0.342 = 0.117$

$Gini\ gain(S, Temperature) = 0.459 - 0.4405 = 0.0185$

$Gini\ gain(S, Humidity) = 0.459 - 0.3674 = 0.0916$

$Gini\ gain(S, windy) = 0.459 - 0.4286 = 0.0304$

Choose one that has a higher Gini gain. Gini gain is higher for outlook. So we can choose it as our root node.

Now you have got an idea of how to proceed further. Repeat the same steps we used in the ID3 algorithm.

Advantages and disadvantages of decision trees

Advantages:

1. Decision trees are super interpretable
2. Require little data preprocessing
3. Suitable for low latency applications

Disadvantages:

1. More likely to overfit noisy data. The probability of overfitting on noise increases as a tree gets deeper. A solution for it is **pruning**. Another way to avoid overfitting is to use bagging techniques like Random Forest.

The Classification and Regression Tree (CART) algorithm is a type of classification algorithm that is required to build a decision tree on the basis of Gini's impurity index.

It is a basic machine learning algorithm and provides a wide variety of use cases. A statistician named Leo Breiman coined the phrase to describe Decision Tree algorithms that may be used for classification or regression predictive modeling issues.

CART is an umbrella word that refers to the following types of decision trees:

- **Classification Trees:** When the target variable is continuous, the tree is used to find the "class" into which the target variable is most likely to fall.
- **Regression trees:** These are used to forecast the value of a continuous variable.

Advantages of CART algorithm

The CART algorithm is nonparametric, thus it does not depend on information from a certain sort of distribution.

1. The CART algorithm combines both testings with a test data set and cross-validation to more precisely measure the goodness of fit.
2. CART allows one to utilize the same variables many times in various regions of the tree. This skill is capable of revealing intricate interdependencies between groups of variables.
3. Outliers in the input variables have no meaningful effect on CART.
4. One can loosen halting restrictions to allow decision trees to overgrow and then trim the tree down to its ideal size. This method reduces the likelihood of missing essential structure in the data set by terminating too soon.
5. To choose the input set of variables, CART can be used in combination with other prediction algorithms.

The CART algorithm is a subpart of Random Forest, which is one of the most powerful algorithms of Machine learning. The CART algorithm is organized as a series of questions, the responses to which decide

the following question if any. The ultimate outcome of these questions is a tree-like structure with terminal nodes when there are no more questions.

Gini's impurity index which measures a distribution among affection of specific-field with the result of instance. It means, it can measure how much every mentioned specification is affecting directly in the resultant case. Gini index is used in the real-life scenario.

Step by Step ID3 Decision Tree Example

Decision tree algorithms transfom raw data to rule based decision making trees. Herein, ID3 is one of the most common decision tree algorithm. Firstly, It was introduced in 1986 and it is acronym of **Iterative Dichotomiser**.

First of all, **dichotomisation means dividing into two completely opposite things**. That's why, the algorithm iteratively divides attributes into two groups which are the most dominant attribute and others to construct a tree. Then, it calculates the entropy and information gains of each attribute. In this way, the most dominant attribute can be founded. After then, the most dominant one is put on the tree as decision node. Thereafter, entropy and gain scores would be calculated again among the other attributes. Thus, the next most dominant attribute is found. Finally, this procedure continues until reaching a decision for that branch. That's why, it is called Iterative Dichotomiser.

No matter which decision tree algorithm you are running: ID3, C4.5, CART, CHAID or Regression Trees. They all look for the feature offering the highest information gain. Then, they add a decision rule for the found feature and build another decision tree for the sub data set recursively until they reached a decision.

Besides, regular decision tree algorithms are designed to create branches for categorical features. Still, we are able to build trees with continuous and numerical features. The trick is here that we will convert continuous features into categorical. We will split the numerical feature where it offers the highest information gain.

ID3 in Python

This blog post mentions the deeply explanation of ID3 algorithm and we will solve a problem step by step. On the other hand, you might just want to run ID3 algorithm and its mathematical background might not attract your attention.

Herein, you can find the *python implementation* of ID3 algorithm here. You can build ID3 decision trees with a few lines of code. This package supports the most common decision tree algorithms such as ID3, C4.5, CART, CHAID or Regression Trees, also some bagging methods such as random forest and some boosting methods such as gradient boosting and adaboost.

Objective

Decision rules will be found based on entropy and information gain pair of features.

Data set

For instance, the following table informs about decision making factors to play tennis at outside for previous 14 days.

Day	Outlook	Temp.	Humidity	Wind	Decision
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

We can summarize the ID3 algorithm as illustrated below

$$Entropy(S) = \sum - p(I) \,.\, \log_2 p(I)$$

$$Gain(S, A) = Entropy(S) - \sum [\, p(S|A) \,.\, Entropy(S|A) \,]$$

These formulas might confuse your mind. Practicing will make it understandable.

Entropy

We need to calculate the entropy first. Decision column consists of 14 instances and includes two labels: yes and no.

There are 9 decisions labeled yes, and 5 decisions labeled no.

$$Entropy(Decision) = - p(Yes) \,.\, \log2p(Yes) - p(No) \,.\, \log2p(No)$$

$$Entropy(Decision) = - (9/14) \,.\, \log2(9/14) - (5/14) \,.\, \log2(5/14) = 0.940$$

Now, we need to find the most dominant factor for decisioning.

Wind factor on decision

Gain(Decision, Wind) = Entropy(Decision) – ∑ [p(Decision | Wind) . Entropy(Decision | Wind)]

Wind attribute has two labels: weak and strong. We would reflect it to the formula.

Gain(Decision, Wind) = Entropy(Decision) – [p(Decision | Wind=Weak) . Entropy(Decision | Wind=Weak)] – [p(Decision | Wind=Strong) . Entropy(Decision | Wind=Strong)]

Now, we need to calculate (Decision | Wind=Weak) and (Decision | Wind=Strong) respectively.

Weak wind factor on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
13	Overcast	Hot	Normal	Weak	Yes

There are 8 instances for weak wind. Decision of 2 items are no and 6 items are yes as illustrated below.

1- Entropy(Decision | Wind=Weak) = – p(No) . log2p(No) – p(Yes) . log2p(Yes)

2- Entropy(Decision | Wind=Weak) = – (2/8) . log2(2/8) – (6/8) . log2(6/8) = 0.811

Notice that if the number of instances of a class were 0 and total number of instances were n, then we need to calculate -(0/n) . log2(0/n). Here, log(0) would be equal to -∞, and we cannot calculate 0 times ∞. This is a special case often appears in decision tree applications. Even though compilers cannot compute this operation, we can compute it with calculus.

Strong wind factor on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
2	Sunny	Hot	High	Strong	No
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
14	Rain	Mild	High	Strong	No

Here, there are 6 instances for strong wind. Decision is divided into two equal parts.

1- Entropy(Decision | Wind=Strong) = – p(No) . log2p(No) – p(Yes) . log2p(Yes)

2- Entropy(Decision | Wind=Strong) = – (3/6) . log2(3/6) – (3/6) . log2(3/6) = 1

Now, we can turn back to Gain(Decision, Wind) equation.

Gain(Decision, Wind) = Entropy(Decision) – [p(Decision | Wind=Weak) . Entropy(Decision | Wind=Weak)] – [p(Decision | Wind=Strong) . Entropy(Decision | Wind=Strong)] = 0.940 – [(8/14) . 0.811] – [(6/14). 1] = 0.048

Calculations for wind column is over. Now, we need to apply same calculations for other columns to find the most dominant factor on decision.

Other factors on decision

We have applied similar calculation on the other columns.

1- Gain(Decision, Outlook) = 0.246

2- Gain(Decision, Temperature) = 0.029

3- Gain(Decision, Humidity) = 0.151

As seen, outlook factor on decision produces the highest score. That's why, outlook decision will appear in the root node of the tree.

Now, we need to test dataset for custom subsets of outlook attribute.

Overcast outlook on decision

Basically, decision will always be yes if outlook were overcast.

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Sunny outlook on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Here, there are 5 instances for sunny outlook. Decision would be probably 3/5 percent no, 2/5 percent yes.

1- Gain(Outlook=Sunny | Temperature) = 0.570

2- Gain(Outlook=Sunny | Humidity) = 0.970

3- Gain(Outlook=Sunny | Wind) = 0.019

Now, humidity is the decision because it produces the highest score if outlook were sunny.

At this point, decision will always be no if humidity were high.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

On the other hand, decision will always be yes if humidity were normal

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Finally, it means that we need to check the humidity and decide if outlook were sunny.

Rain outlook on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

1- Gain(Outlook=Rain | Temperature) = 0.01997309402197489

2- Gain(Outlook=Rain | Humidity) = 0.01997309402197489

3- Gain(Outlook=Rain | Wind) = 0.9709505944546686

Here, wind produces the highest score if outlook were rain. That's why, we need to check wind attribute in 2nd level if outlook were rain.

So, it is revealed that decision will always be yes if wind were weak and outlook were rain.

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

What's more, decision will be always no if wind were strong and outlook were rain.

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

So, decision tree construction is over.

Feature Importance

Decision trees are naturally explainable and interpretable algorithms. Besides, we can find the feature importance values as well to understand how model works.

References:

1. https://sefiks.com/2017/11/20/a-step-by-step-id3-decision-tree-example/
2. https://medium.datadriveninvestor.com/decision-tree-algorithm-with-hands-on-example-e6c2afb40d38
3. https://www.saedsayad.com/decision_tree.htm