Decision trees

- A decision tree is a tree where each node represents a feature (attribute), each link (branch) represents a decision (rule) and each leaf represents an outcome (categorical or continue value).
- Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas.
- In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions.
- It is one of the most widely used and practical methods for supervised learning.
- Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks.
- The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

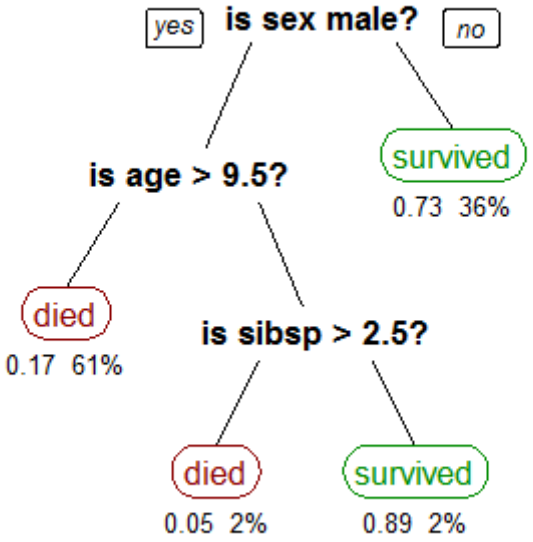
From https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/

Decision trees

Example: Titanic data set for predicting whether a passenger will survive or not.

The model uses 3 features/attributes/columns from the data set, namely sex, age and sibsp (number of siblings or spouses aboard).

The leaf nodes of the tree contain an output variable (y) which is used to make a prediction. Predictions are made by walking the splits of the tree until arriving at a leaf node and output the class value at that leaf node.



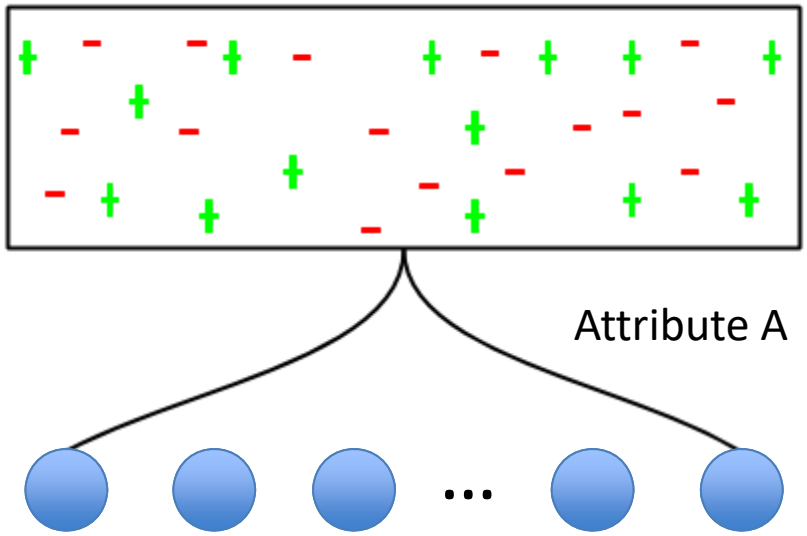
Probability of survival
Percentage of observations in the leaf

From Wikipedia

Decision trees

The steps to the algorithm to create a decision tree are:

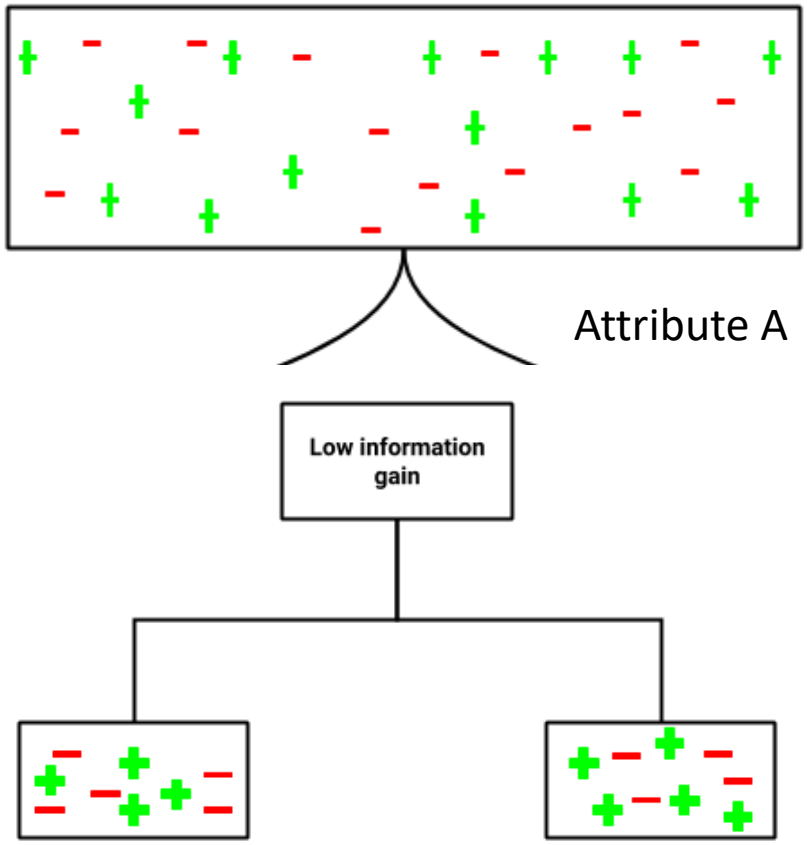
1. Select the best attribute → A
2. Assign A as the decision attribute (test case) for the NODE
3. For each value of A, create a new descendant of the NODE
4. Sort the training examples to the appropriate descendant node leaf
5. If examples are perfectly classified, then STOP else iterate over the new leaf nodes.



Decision trees

The steps to the algorithm to create a decision tree are:

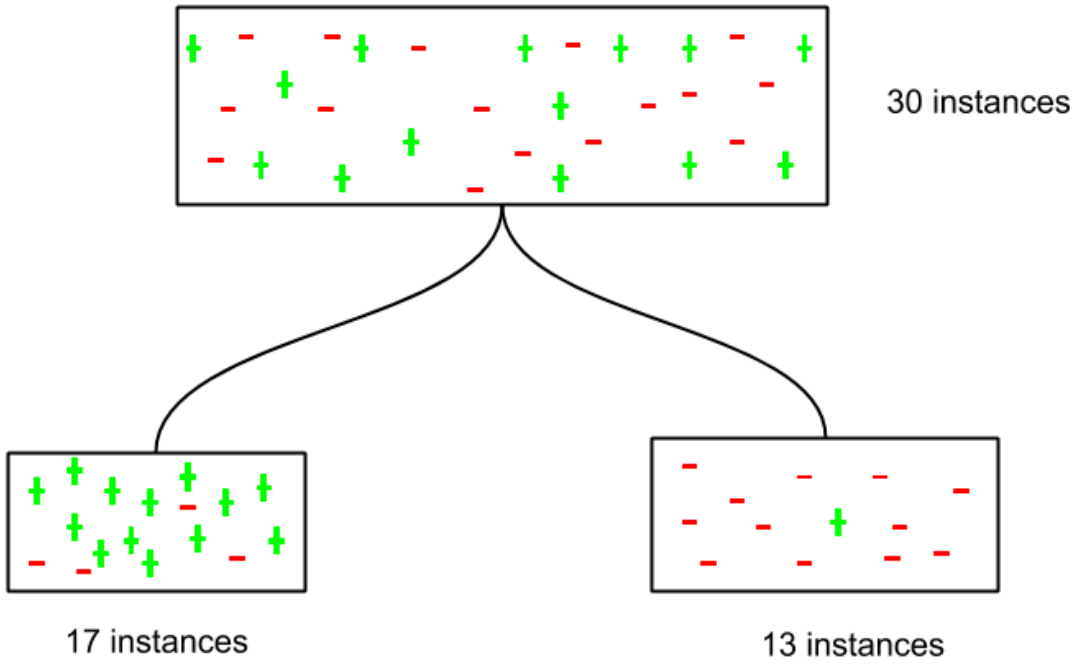
1. Select the best attribute → A
2. Assign A as the decision attribute (test case) for the NODE
3. For each value of A, create a new descendant of the NODE
4. Sort the training examples to the appropriate descendant node leaf
5. If examples are perfectly classified, then STOP else iterate over the new leaf nodes.



Decision trees

The steps to the algorithm to create a decision tree are:

1. Select the best attribute → A
2. Assign A as the decision attribute (test case) for the NODE
3. For each value of A, create a new descendant of the NODE
4. Sort the training examples to the appropriate descendant node leaf
5. If examples are perfectly classified, then STOP; otherwise, iterate over the new leaf nodes.

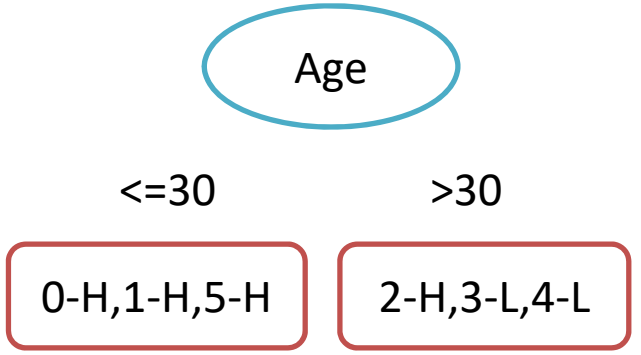


Decision trees: Example

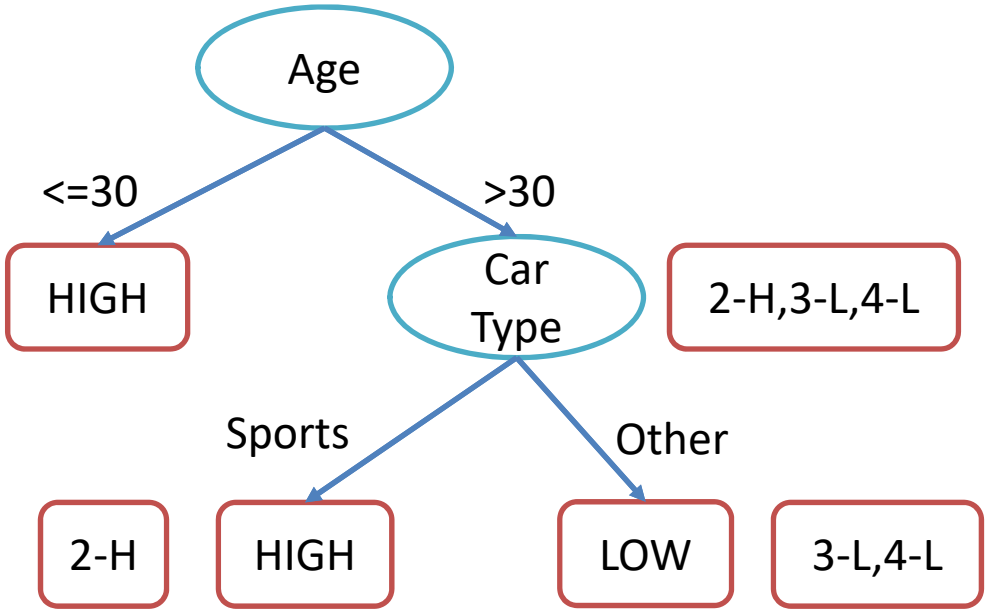
The steps to the algorithm to create a decision tree are:

1. Select the best attribute → A
2. Assign A as the decision attribute (test case) for the NODE
3. For each value of A, create a new descendant of the NODE
4. Sort the training examples to the appropriate descendant node leaf
5. If examples are perfectly classified, then STOP; otherwise, iterate over the new leaf nodes.

Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High



Decision trees: Example



Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

Decision trees: Simple exercise

Sleepy	Hungry	Good mood	Productive?
No	No	No	No
No	No	Yes	Yes
No	Yes	No	No
No	Yes	Yes	Yes
Yes	No	No	No
Yes	No	Yes	Yes
Yes	Yes	No	No
Yes	Yes	Yes	No

Decision trees: Impurity measure → Gini index

It is a measure of how 'diverse' the data in a node is. If the data consisted of only one class, the gini index would be 0. Thus at each node the gini index is used to find a split which makes the data less diverse.

The gini diversity index at any node is given by: $GINI(t) = 1 - \sum_{i=1}^{K} p(c_i|t)^2$

Where K is the number of classes, c_i is each of that classes and t is a given node.

For example:

GINI(root)=1 – ((4/6)^2 + (2/6)^2) = 1 – (0.444 + 0.111) = 0.445

Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

The Gini index goes through:

- 0 when all samples belong to one class, which is the best situation (no diversity)
- and 1-1/K, when samples are equally distributed among all classes, which is the worst situation (maximum diversity)

Decision trees: Impurity measure → Gini index

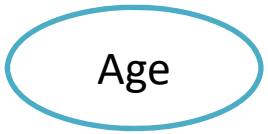
To find the goodness of fit for an split at a particular node p, we need to take into account the probabilities of the classes coming into the node, and the probabilities of the B outcomes after the split has been applied (which in a decision tree is the number of branches coming from a node). So, we define a term called GINI(split) as:

$$GINI(split) = \sum_{i=1}^{B} \frac{n_i}{n} GINI(i)$$

Where n_i is the number of samples at child i and n is the number of samples at node p.

The split for a node is chosen by minimising the GINI function.

Decision trees: Example of Gini index



<=30?

	<=30	>30
High	3	1
Low	0	2

Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

GINI(<=30) = $1 - ((3/3)^2 + (0/3)^2) = 0$

GINI(>30) = $1 - ((1/3)^2 + (2/3)^2) = 0.445$

GINI(split1) = $(3/6 * 0 + 3/6 * 0.445) = 0.222$

Decision trees: Example of Gini index

Age

<=40?

	<=40	>40
High	3	1
Low	1	1

Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

Age

<=50?

	<=50	>50
High	4	0
Low	1	1

$GINI(<=40) = 1 - ((3/4)^2 + (1/4)^2) = 0.375$

$GINI(>40) = 1 - ((1/2)^2 + (1/2)^2) = 0.5$

$GINI(split2) = (4/6 * 0.375 + 2/6 * 0.5) = 0.416$

$GINI(<=50) = 1 - ((4/5)^2 + (1/5)^2) = 0.32$

$GINI(>50) = 1 - ((0/1)^2 + (1/1)^2) = 0$

$GINI(split2) = (5/6 * 0.32 + 1/6 * 0) = 0.26$

Decision trees: Exercise of Gini index

Given the following dataset, where the variable to predict is "profitable", calculate the GINI index of the following splits, and tell which one you would choose and why.

a. Price, two branches: {low, med} and {high}
b. Maintenance: two branches: {high, med} and {low}
c. Capacity two branches: {2} and {4,5}

price	maintenance	capacity	airbag	profitable
low	low	2	no	yes
low	med	4	yes	no
low	low	4	no	yes
low	high	4	no	no
med	med	4	no	no
med	med	4	yes	yes
med	high	2	yes	no
med	high	5	no	yes
high	med	4	yes	yes
high	high	2	yes	no
high	high	5	yes	yes

Decision trees: Advantages

- Simple to understand and to interpret. Trees can be visualized.
- It requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed.
- The cost of using the tree (i.e., predicting data) is low.
- It is able to handle both numerical and categorical data.
- It uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.

Decision trees: Disadvantages

- Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting. Some hyperparameters tunning mechanisms may be necessary to avoid this problem.
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.
- Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.