Introduction

A Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

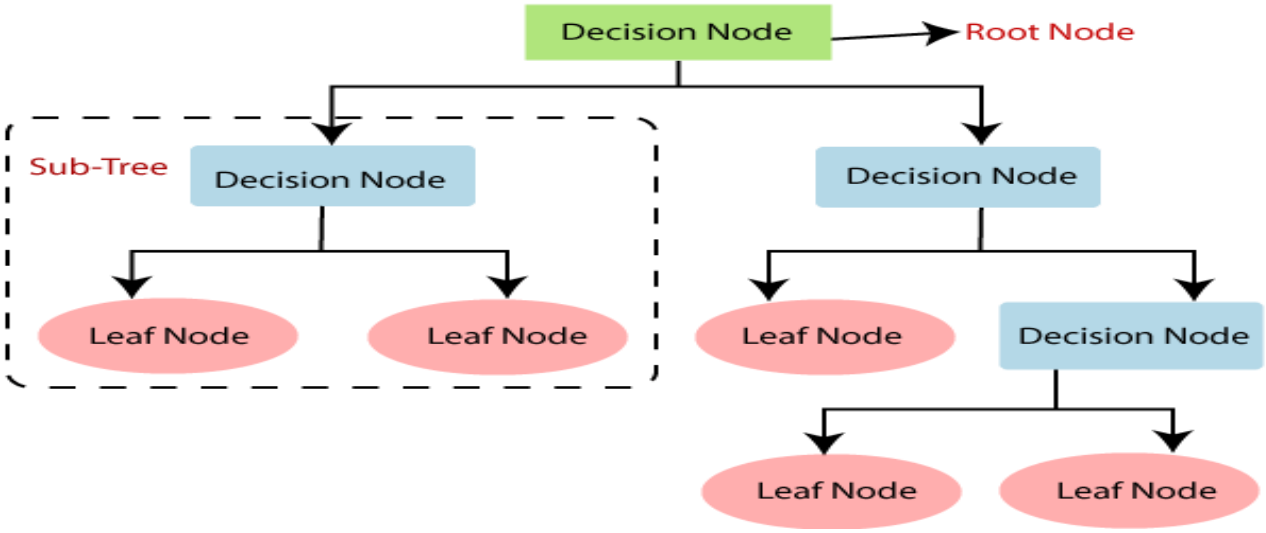
In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

To build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.



Decision Tree Terminologies:

Root Node: The root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Branch/Sub Tree: A tree formed by splitting the tree.

Pruning: Pruning is the process of removing unwanted branches from the tree.

Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

Python Implementation of Decision Tree.

Steps:

1. Data Pre-processing step
2. Fitting a Decision-Tree algorithm to the Training set
3. Predicting the test result
4. Test accuracy of the result(Creation of Confusion matrix)
5. Visualizing the test set result.

DecisionTreeID3

Implement ID3 algorithm for data with all categorical attributes by using panda and numpy (sklearn DecisionTreeClassifier doesn't support categorical attributes).

Stopping criteria

max_depth: the max depth of the tree.
min_samples_split: the minimum number of samples in a split to be considered.
min_gain: minimum gain for splitting.a

simple test on weather.csv

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rainy	mild	high	strong	no

Result should be:

```
['no', 'no', 'yes', 'yes', 'yes', 'no', 'yes', 'no', 'yes', 'yes', 'yes', 'yes', 'yes', 'no']
```

It means 100% accuracy on training set.

File used: weather.csv

References

1. C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning): J. Ross Quinlan
2. Kaggle
3.