Decission Trees

DESU IA4Health

Info-GRAM

SESSTIM, Faculté des Sciences Médicales et Paramédicales, Aix-Marseille Université, Marseille, France
http://sesstim.univ-amu.fr/

Characteristics

- *Decision Trees* can perform both classification and regression tasks, and even multioutput tasks.
- Decision Trees are also the fundamental components of Random Forests.
- Non-parametrics models
- they don’t require feature scaling or centering at all.
- Decision Trees are fairly intuitive and their decisions are easy to interpret. They are often called *white box models*. In contrast, as we will see, Random Forests or neural networks are generally considered *black box models*.

Objectives

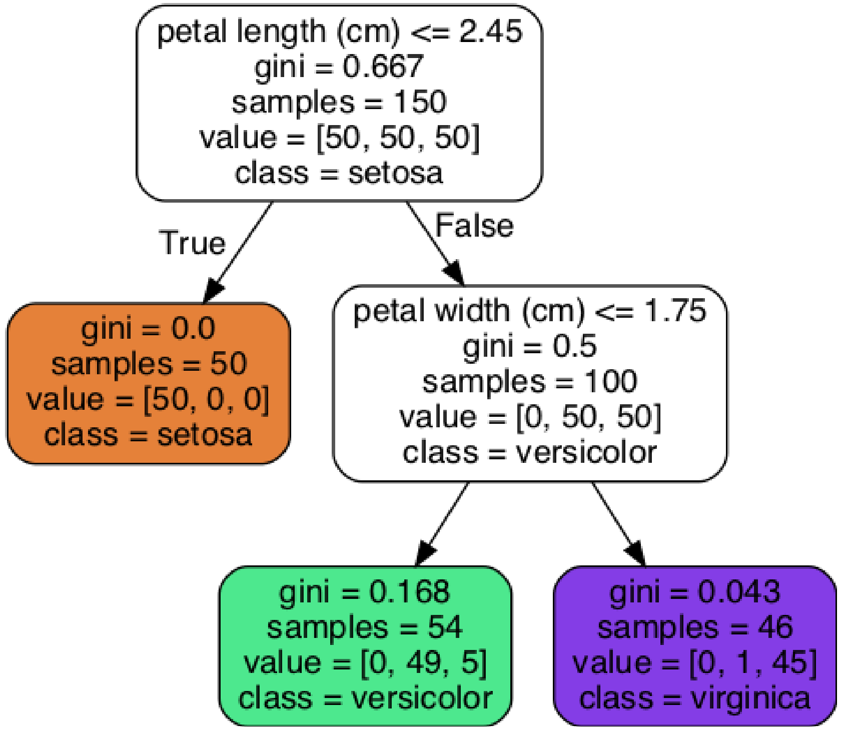
- Trainning
- Trees visualization
- Make predictions
- Regularization
- Trees for regression

Tree representation

- **Samples :** counts how many training instances it applies to.
- **Value :** tells you how many training instances of each class this node applies
- **Gini :** measures its *impurity*:
  - a node is "pure" (gini=0) if all training instances it applies to belong to the same class.

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

$p_{i,k}$ is the ratio of class k instances among the training instances in the i^{th} node.

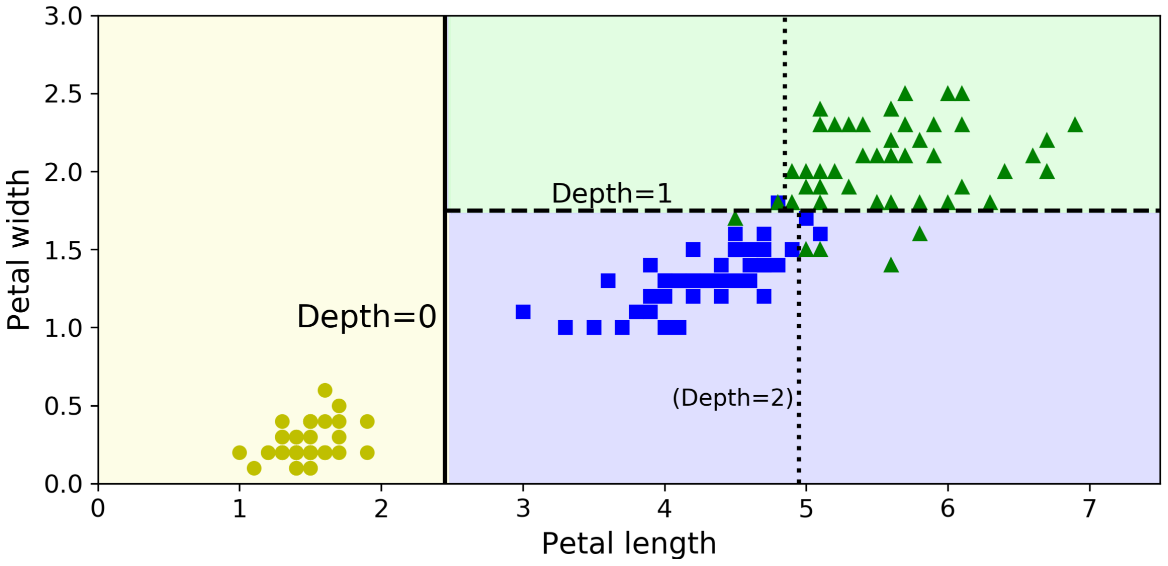


Entropy

- The concept of entropy originated in thermodynamics as a measure of molecular disorder: entropy approaches zero when molecules are still and well ordered.
- Shannon's *information theory*, where it measures the average information content of a message: entropy is zero when all messages are identical.

Decision Boundaries

- The thick vertical line represents the decision boundary of the root node (depth 0): petal length = 2.45 cm. Since the left area is pure (only Iris-Setosa), it cannot be split any further.
- The right area is impure, so the depth-1 right node splits it at petal width = 1.75 cm (represented by the dashed line). Since max_depth was set to 2, the Decision Tree stops right there.
- if max_depth would have been 3, then the two depth-2 nodes would each add another decision boundary (represented by the dotted lines).



Estimating Class Probabilities

- A Decision Tree can also estimate the probability that an instance belongs to a particular class *k*: first it traverses the tree to find the leaf node for this instance, and then it returns the ratio of training instances of class *k* in this node

CART Training Algorithm

- The algorithm first splits the training set in two subsets using a single feature k and a threshold t_k (e.g., "petal length ≤ 2.45 cm").
  - It searches for the pair (k, t_k) that produces the purest subsets (weighted by their size).
- The cost function that the algorithm tries to minimize

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

$$\text{where } \begin{cases} G_{\text{left/right}} \text{ measures the impurity of the left/right subset,} \\ m_{\text{left/right}} \text{ is the number of instances in the left/right subset.} \end{cases}$$

- Once it has successfully split the training set in two, it splits the subsets, recursively. It stops recursing once it reaches the maximum depth (defined by the max_depth hyperparameter),

Regularization Hyperparameters

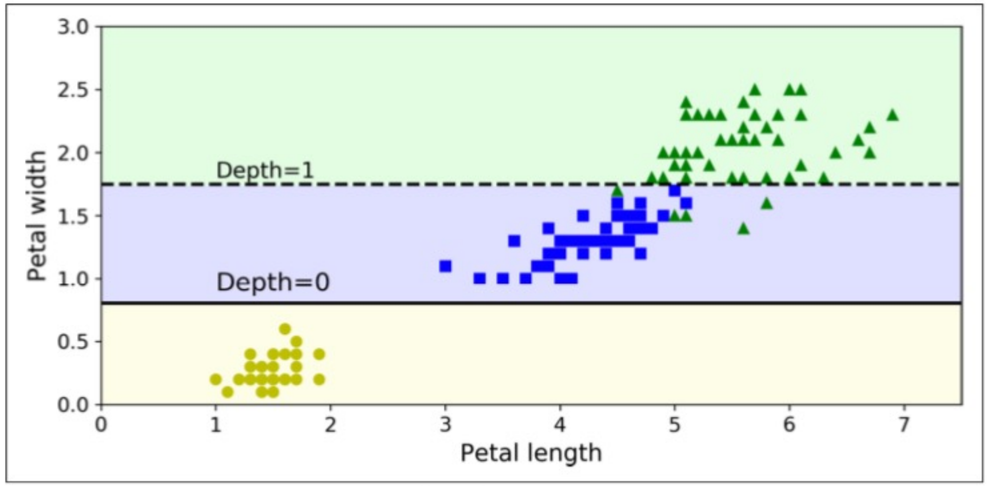
- ***max_depth***
- ***max_leaf_nodes*** (maximum number of leaf nodes),
- ***max_features*** (maximum number of features that are evaluated for splitting at each node).
- ***min_samples_split*** (the minimum number of samples a node must have before it can be split),
- ***min_samples_leaf*** (the minimum number of samples a leaf node must have),
- ***min_weight_fraction_leaf*** (same as min_samples_leaf but expressed as a fraction of the total number of weighted instances)

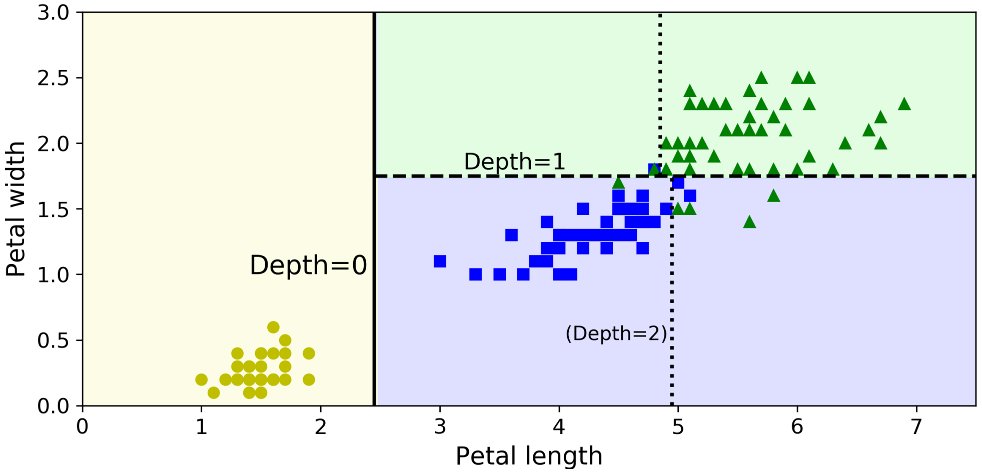
To regularize the model :

- Increase min_* hyperparameters
- Reduce max_*hyperparameters

Inestability

- Decision Trees are very sensitive to small variations in the training data.
- For example, if you just remove the widest Iris-Versicolor from the iris training set (the one with petals 4.8 cm long and 1.8 cm wide) and train a new Decision Tree, you may get the model represented in the left very different from the previous Decision Tree in the right





Solution :

- Ensemble classifiers