Machine Learning Engineer Course
Day 13

- Decision Tree -

DIVE INTO CODE

Thursday June 3, 2021

DIOP Mouhamed

Agenda

1. Check-in
2. How to proceed
3. Quick Review
4. Decision Tree
5. Tree module of Scikit-learn
6. Assignment
7. Scratch Decision Tree – Sample Code
8. Check-out

Check-in

3 minutes Please post the following point to Zoom chat.

Q. What is your main purpose in this course (your goal) ?

(Anything is fine.)

How to proceed - Objective

What is the purpose?

1. Understanding Decision Tree though scratch

2. Getting familiar with the implementation of complex algorithms

Let’s learn the basics of Decision Tree

Quick Review (Scratch SVM)

Knowing the SVM Problem Setting

① Formulate an equation (hypothetical function) to derive predictions

② Set up a problem to be minimized and formulate an equation (objective function).

③ Replace the objective function of the main problem with the objective function of the dual problem

④ Exploratory search for the optimal solution of the objective function (gradient method)

⑤ Estimate using the obtained sparse solution (support vector)

What is Decision tree?

A method of obtaining a hyperplane (discrimination boundary) that is parallel to the real value features by combining simple discriminating rules. Specifically, it is a tree-structured representation of the process of determining the magnitude relationship between a feature value and a threshold.

What are the given conditions ?

The following is assumed in the decision tree:

① Do not assume the distribution of the data to be analyzed
② It is composed of multiple step functions (1)

(1) A function that returns 1 if the input is greater than a certain value, otherwise 0.

Target audience for this assignment

① Those who can write code to learn and estimate using scikit-learn classification model.

② Those who have solved the Sprint5 Scratch SVM

Decision Tree – The Flow

Knowing the Decision Tree Problem Setting

① Use an arbitrary value of the feature value as the "threshold" for segmentation (done for each feature value)

② Divide the sample by the threshold value in ① above (done for each feature)

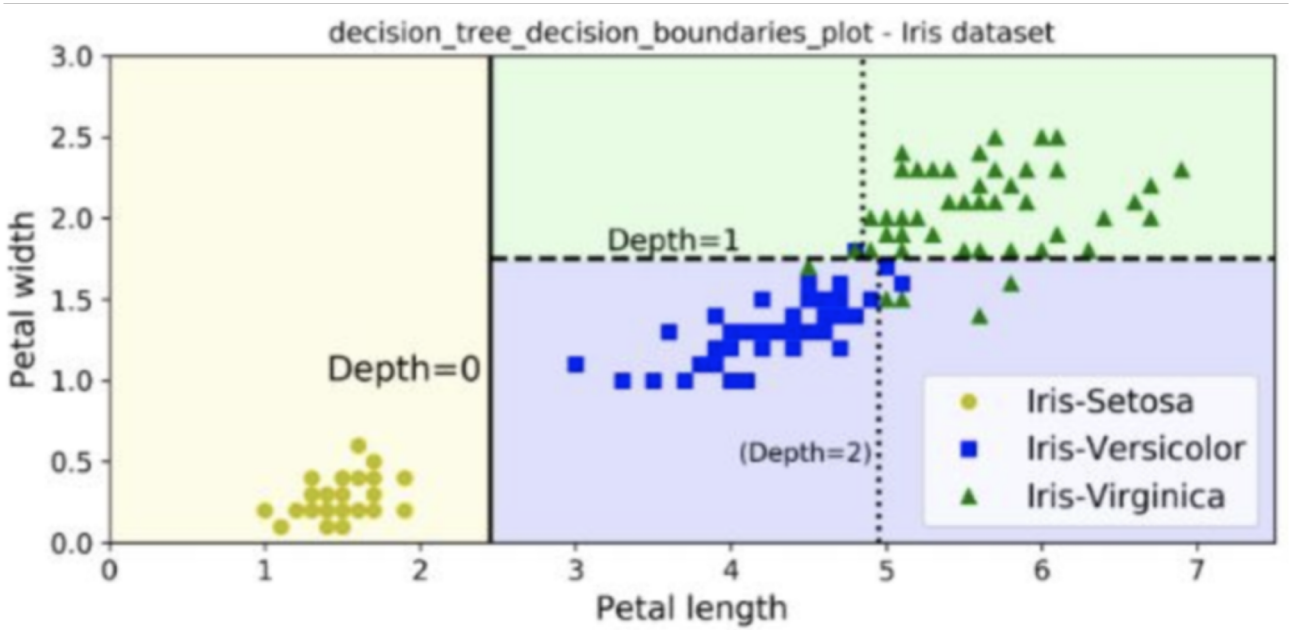
③ Find the difference between the sum of the gini impurity of the samples in each group after the split and the gini impurity of all samples before the split (calculate the information gain).

④ The one that maximizes the difference (information gain) is used as the decision criterion for splitting the root node.

Premise

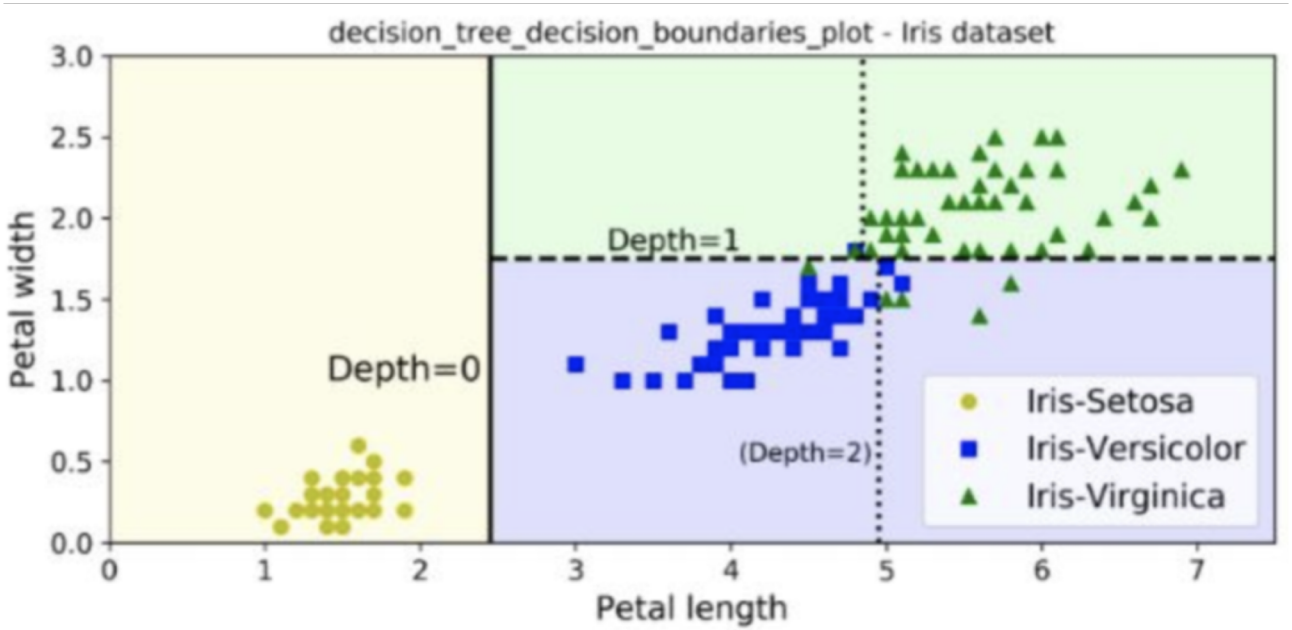
Iris data

Let’s say you have an iris dataset here. The data points are color-coded for each class in advance. Let’s select a feature X1 (petal length) and a feature X2 (petal width) and plot the relationship system between two variables



Premise

This time, it would be nice if we could draw an identification boundary (boundary line) for classifying the classes of Iris-setosa, Iris-versicolor and Iris-verginica. It is said that the decision tree repeats division at the decision boundary parallel to the axis, but how do you select the division point ?



Decision Tree – The Flow

Knowing the Decision Tree Problem Setting

① Use an arbitrary value of the feature value as the "threshold" for segmentation (done for each feature value)

② Divide the sample by the threshold value in ① above (done for each feature)

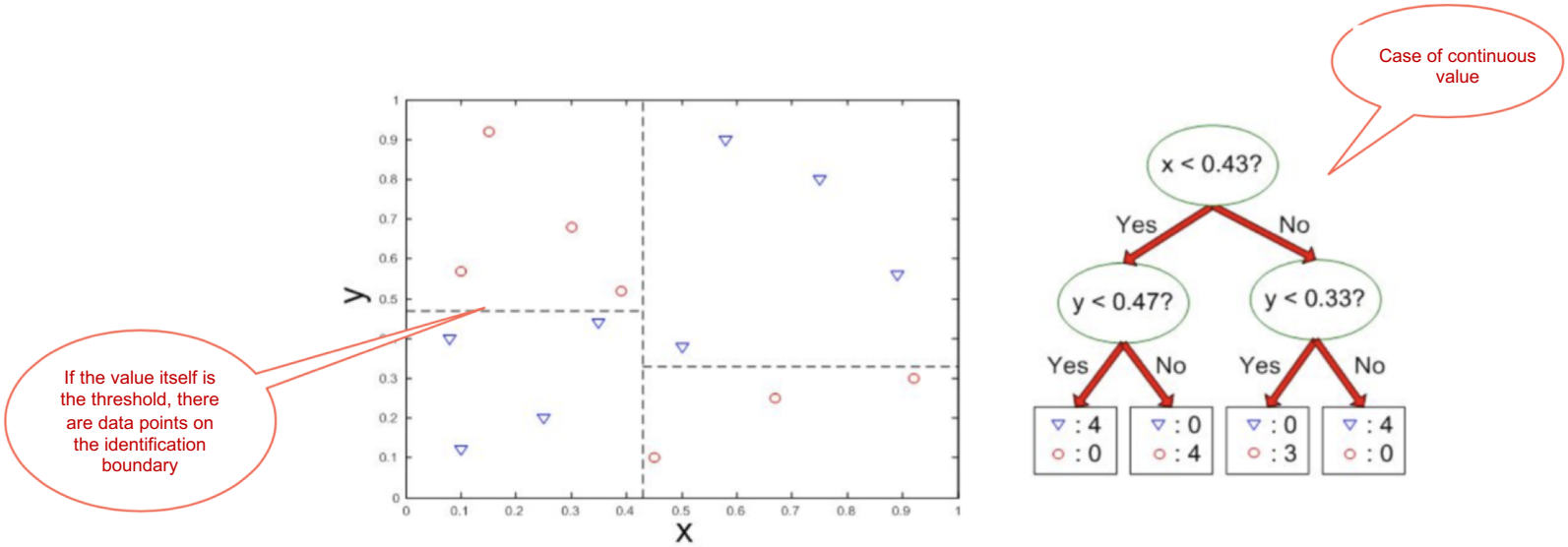
③ Find the difference between the sum of the gini impurity of the samples in each group after the split and the gini impurity of all samples before the split (calculate the information gain).

④ The one that maximizes the difference (information gain) is used as the decision criterion for splitting the root node.

Division and algorithm

The division rule

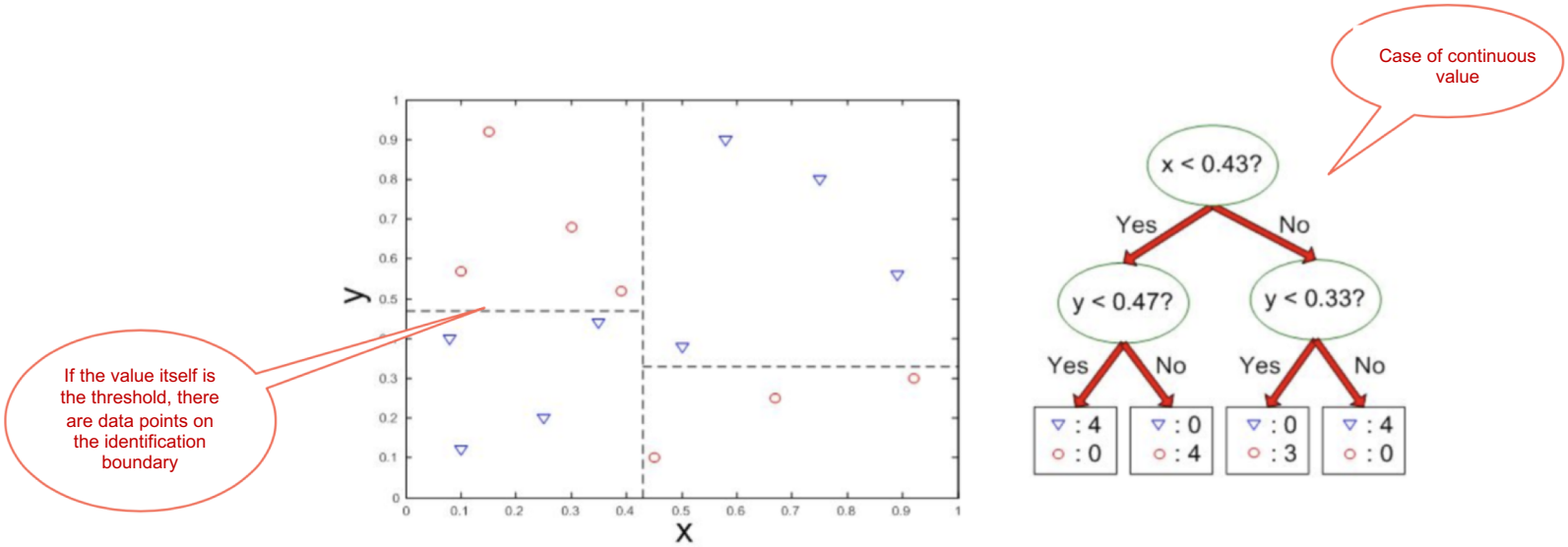
Decision tree is selected by by evaluating the division candidate points with a certain division index. At this time, the identification boundary is orthogonal to the feature axis of the d-dimensional space.



Division and algorithm

When the feature quantity is a continuous value
when the number of training data is n, there are n-1 discrete division candidate points.

When the feature quantity is a nominal scale/ordinal scale
there are exactly as many division candidate points as the number of categories.

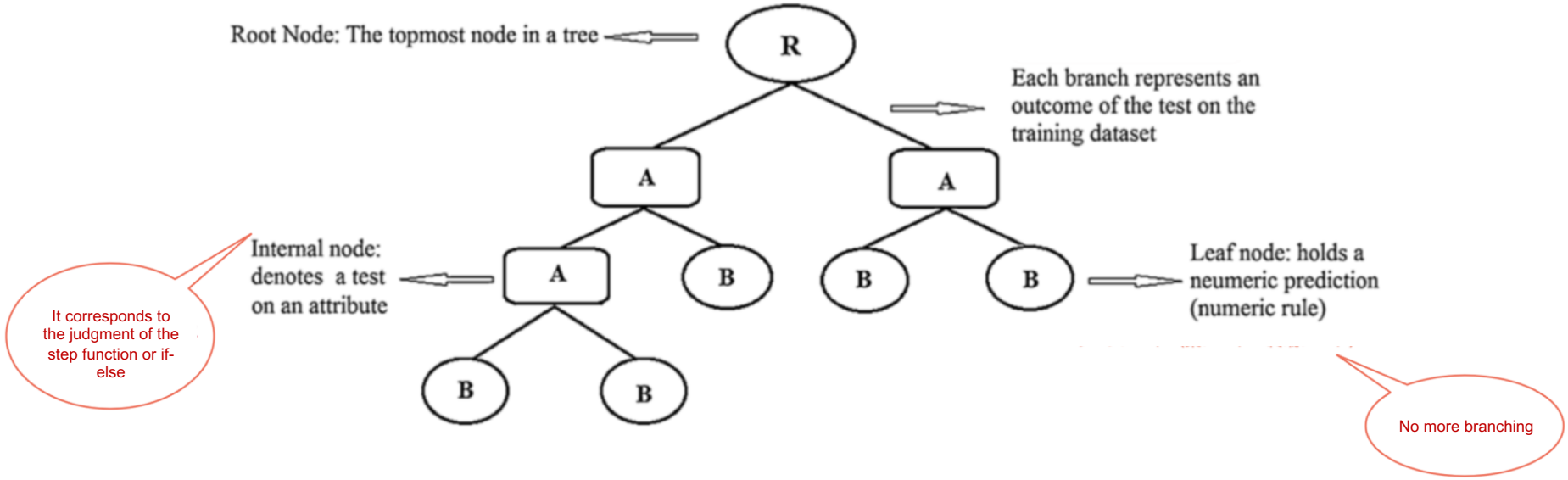


Division and algorithm

Three structure algorithm
This time we can use the algorithm called **CART (classification and regression tree)**, which is a binary tree.

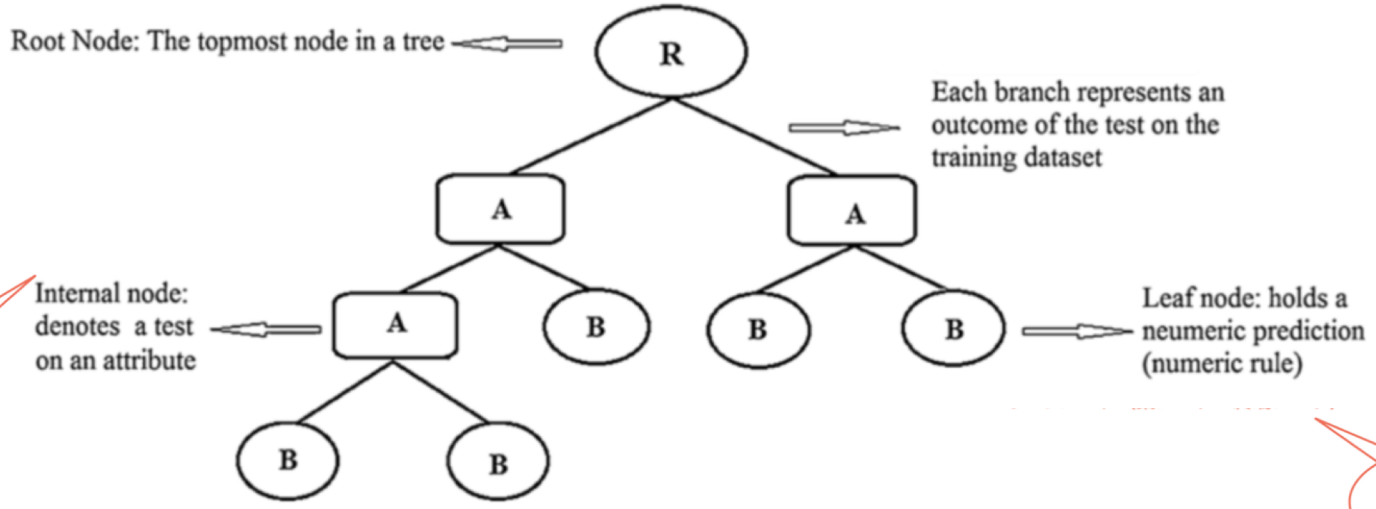
The algorithm supports both classification and regression.

(1) There is also another algorithm called C4.5 that generates an N-branch tree, This is only applicable to classification problems.



Division and algorithm

Each node represents the judgement of the attribute (feature name), each link represents the branch, and each leaf represents the result (categorical value or continuous value)



Decision Tree – The Flow

Knowing the Decision Tree Problem Setting

① Use an arbitrary value of the feature value as the "threshold" for segmentation (done for each feature value)

② Divide the sample by the threshold value in ① above (done for each feature)

③ Find the difference between the sum of the gini impurity of the samples in each group after the split and the gini impurity of all samples before the split (calculate the information gain).

④ The one that maximizes the difference (information gain) is used as the decision criterion for splitting the root node.

Objective Function（Gini impurity）

Objective Function (in the case of Classification)

Gini impurity (also known as the Gini index) or cross-entropy is used as the division index for the classification model of CART. This time, the division is performed based on the Gini impurity (error rate at node t). The Gini impurity is formulated as follows.

Objective Function（Gini impurity）

When considering a trial in which a sample taken from a node is set to 1 when it is the i-th class and 0 when it is the other class (this is called Bernoulli trial), it is the Gini impurity. It can be said that there is a probability that the sample classes at the node are different (there are almost the same number of samples of class 1 and class 0), that is, the bias is small). It also corresponds to the sum of the variances of all classes in the Bernoulli distribution.

t is the node index

i is class index

K is the number of classes

C_i is the i-th class

P(C_ilt) is the percentage of the C_i on the t-th class

$$I(t) = 1 - \sum_{i=1}^{K} P^2(C_i|t) = 1 - \sum_{i=1}^{K} \left(\frac{N_{t,i}}{N_{t,al}} \right)^2$$

$$= \sum_{i=1}^{K} P(C_i|t)(1 - P(C_i|t))$$

N_t,i is the number of samples belonging to the i-th class of the t-th node

N_t,all is the total number of samples for the t-th node

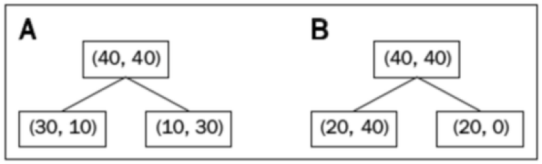
Meaning of the formula:

The calculation of multiplying the probability P that “class” is selected by node t and the probability 1-P that other than “class” is selected is performed for K classes and added.

Objective function (Gini impurity and information gain)

Let's check how to calculate Gini impurity by hand

The figure on the right shows two decision trees (A and B) showing the number of classes before and after division. Let's calculate the Gini impurity in each of A and B.



Click here for other classes

https://www.randpy.tokyo/entry/decision_tree_theory

$$1 - \sum_{i=1}^{K} \left(\frac{N_{t,i}}{N_{t,al}} \right)^2$$

Root node

$$I_G(D_p) = 1 - \left(\left(\frac{40}{80} \right)^2 + \left(\frac{40}{80} \right)^2 \right) = 1 - (0.5^2 + 0.5^2) = 0.5$$

$$A : I_G(D_{left}) = 1 - \left(\left(\frac{30}{40} \right)^2 + \left(\frac{10}{40} \right)^2 \right) = 1 - \left(\frac{9}{16} + \frac{1}{16} \right) = \frac{3}{8} = 0.375$$

$$A : I_G(D_{right}) = 1 - \left(\left(\frac{10}{40} \right)^2 + \left(\frac{30}{40} \right)^2 \right) = 1 - \left(\frac{1}{16} + \frac{9}{16} \right) = \frac{3}{8} = 0.375$$

$$IG(p) = I(p) - \frac{N_{left,all}}{N_{p,all}} I(left) - \frac{N_{right,all}}{N_{p,all}} I(right)$$

$$A : I_G = 0.5 - \frac{40}{80} \times 0.375 - \frac{40}{80} \times 0.375 = 0.125$$

The last part is the calculation of information gain (evaluate the difference before and after decision. The larger the better).

$$B : I_G(D_{left}) = 1 - \left(\left(\frac{20}{60} \right)^2 + \left(\frac{40}{60} \right)^2 \right) = 1 - \left(\frac{9}{16} + \frac{1}{16} \right) = 1 - \frac{5}{9} = 0.44$$

$$B : I_G(D_{right}) = 1 - \left(\left(\frac{20}{20} \right)^2 + \left(\frac{0}{20} \right)^2 \right) = 1 - (1 + 0) = 1 - 1 = 0$$

This one too

$$B : I_G = 0.5 - \frac{60}{80} \times 0.44 - 0 = 0.5 - 0.33 = 0.17$$

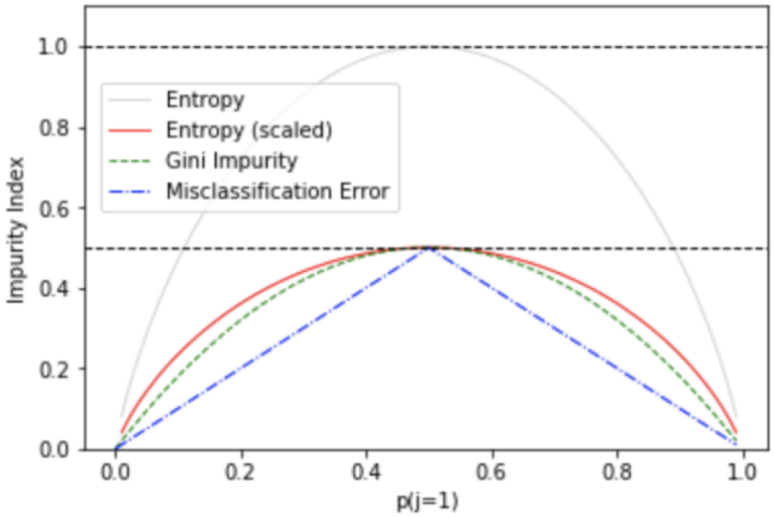
Decision Tree – The Flow

Possible range of Gini impurity.

The relationship between the probability that class j is selected at node t (horizontal axis) and the purity index (vertical axis) can be represented by the following graph. When the class probability is 0.5, the Gini impurity has a maximum value of 0.5. When completely split, the purity is 0.

Click here for the code in the graph below

https://www.bogotobogo.com/python/scikit-learn/scikt_machine_learning_Decision_Tree_Learning_Informatioin_Gain_IG_Impurity_Entropy_Gini_Classification_Error.php



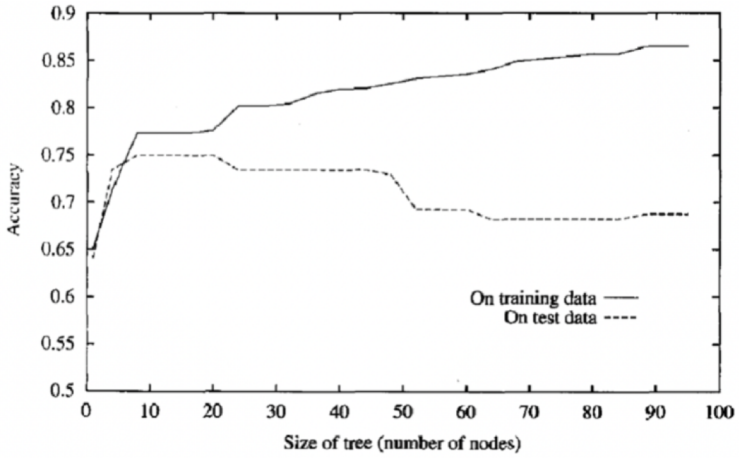
The smaller the purity the better

Decision Tree – The Flow

Problems with decision trees.

The more complex the model is, the more likely it is to overfit the training data. Then, if the training data obtained is significantly different, the tree structure obtained after learning will also be significantly different. At this time, the model is said to have high variance.

There is a method called bagging, in which a sample is randomly re-extracted (put strap sample) from the training data, a decision tree is applied each, and a majority vote is made for the results of multiple decision trees. It consists of the idea of compensating for the instability of the results from a single decision tree.



Decision Tree – The Flow

When do you stop branching ?

If there are many features, many divisions will occur, resulting in a huge tree. Such trees are complex and can lead to overfitting

One way to avoid overfitting is to set a **minimum number** of input data to use on each leaf.

Another method is "**pruning**", which deletes **less important** judgments. This avoids the effects of outliers and prevents overfitting.

Decision Tree – The Flow

Knowing the Decision Tree Problem Setting

① Use an arbitrary value of the feature value as the "threshold" for segmentation (done for each feature value)

② Divide the sample by the threshold value in ① above (done for each feature)

③ Find the difference between the sum of the gini impurity of the samples in each group after the split and the gini impurity of all samples before the split (calculate the information gain).

④ The one that maximizes the difference (information gain) is used as the decision criterion for splitting the root node.

Decision Tree Completed

Tree module of scikit-learn

Let’s first have a look at the one used until now with the help of the scikit-learn library.

Scikit-learn’s Tree module

Sprint 6 – Scratch Decision Tree

Explanation about this Sprint is given but please try it on your own first.

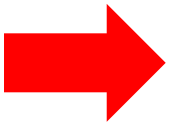
Sprint 6 – Decision Tree

Please work on your own after class and submit your assignments on DIVER.

Sprint 6 – Sample Code

A Sample Code of this Sprint is given but please try it on your own.

Sprint 6 – Decision Tree



Please work on your own after class and submit your assignments on DIVER.

ToDo by next class

Next class will be Zoom : Thursday June 10, 2021 19:30~20:30

🚨ToDo: Scratch Clustering
https://diver.diveintocode.jp/curriculums/1649

Check-out

3 minutes Please post the following point to Zoom chat.

Q. Current feelings and reflections
(joy, anger, sorrow, anticipation, nervousness, etc.)

Thank You For Your Attention