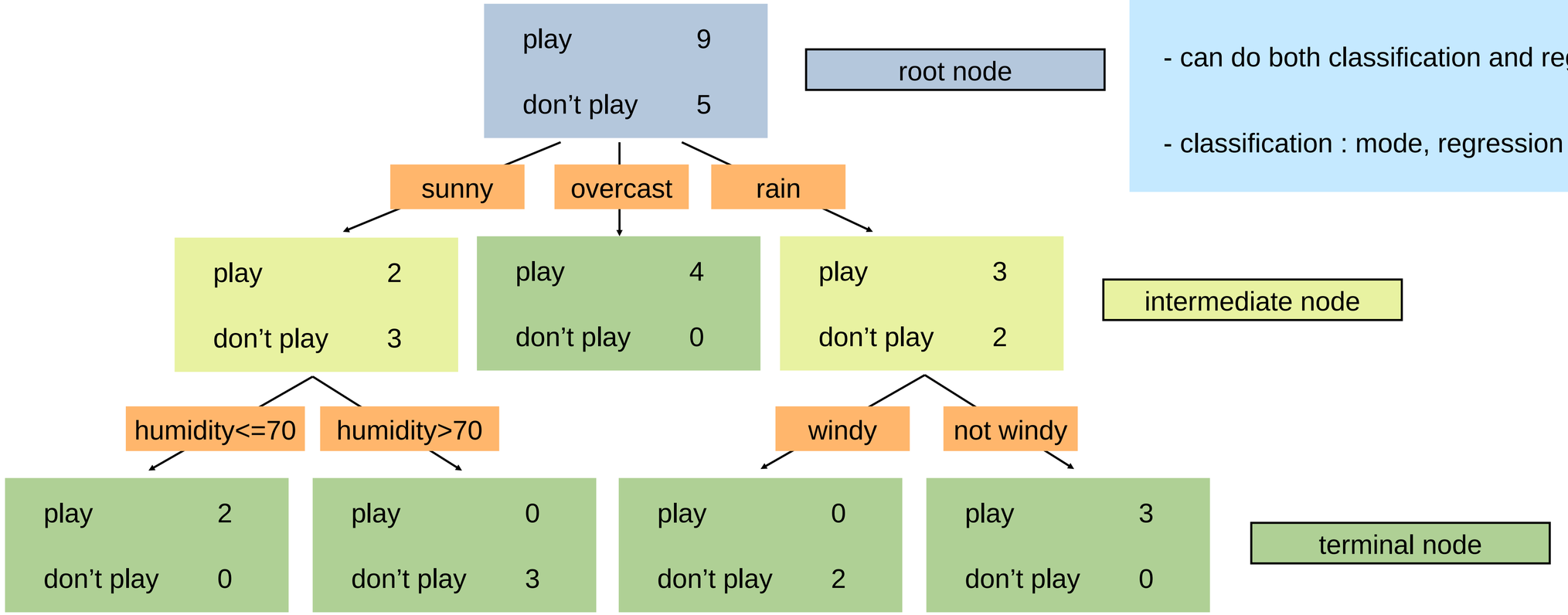
Decision Tree

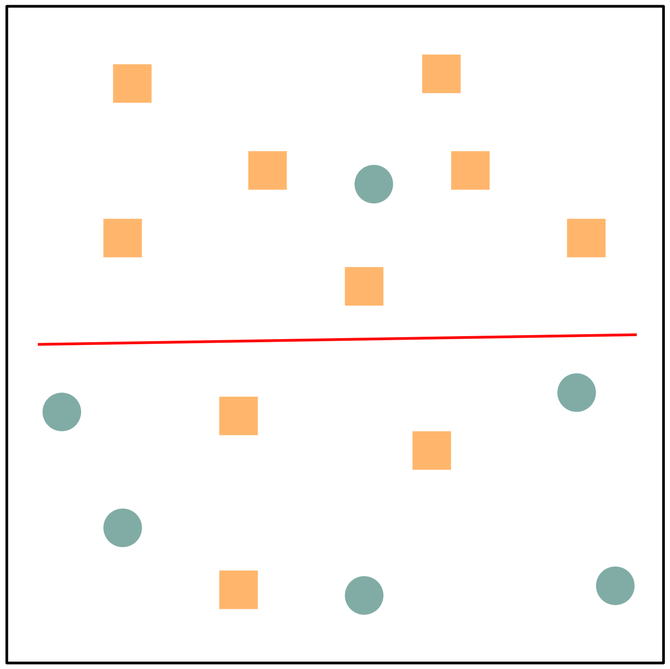
e.g. dependent variable : play?



- number of terminal nodes' data = number of root node's data
  (no intersection between terminal nodes)
- number of terminal nodes = number of subsets
- can do both classification and regression
- classification : mode, regression : mean

Decision Tree

classification



A

we want to...

- maximize homogeneity
- minimize impurity (uncertainty)

→ **“information gain”**

before classification

number of data type : m=2

$$Entropy(A) = -\sum_{k=1}^{m} p_k \log_2 p_k$$

p_k : ratio between category k data and data in area A

$$Entropy(A) = -\frac{10}{16}\log_2\frac{10}{16} - \frac{6}{16}\log_2\frac{6}{16} \approx 0.95$$

after classification : divide to 2 subsets R_1, R_2

$$Entropy(A) = \sum_{i=1}^{d} R_i \left[-\sum_{k=1}^{m} p_k \log_2 p_k \right]$$

R_i : percentage of before dividing data that belongs to i after dividing

$$Entropy(A) = 0.5 \times \left(-\frac{7}{8}\log_2\frac{7}{8} - \frac{1}{8}\log_2\frac{1}{8} \right)$$

$$+ 0.5 \times \left(-\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8} \right) \approx 0.75$$

decreased

Decision Tree

impurity has minimum (homogeneity has maximum), entropy is 0

impurity has maximum (homogeneity has minimum), entropy is 1

Gini index

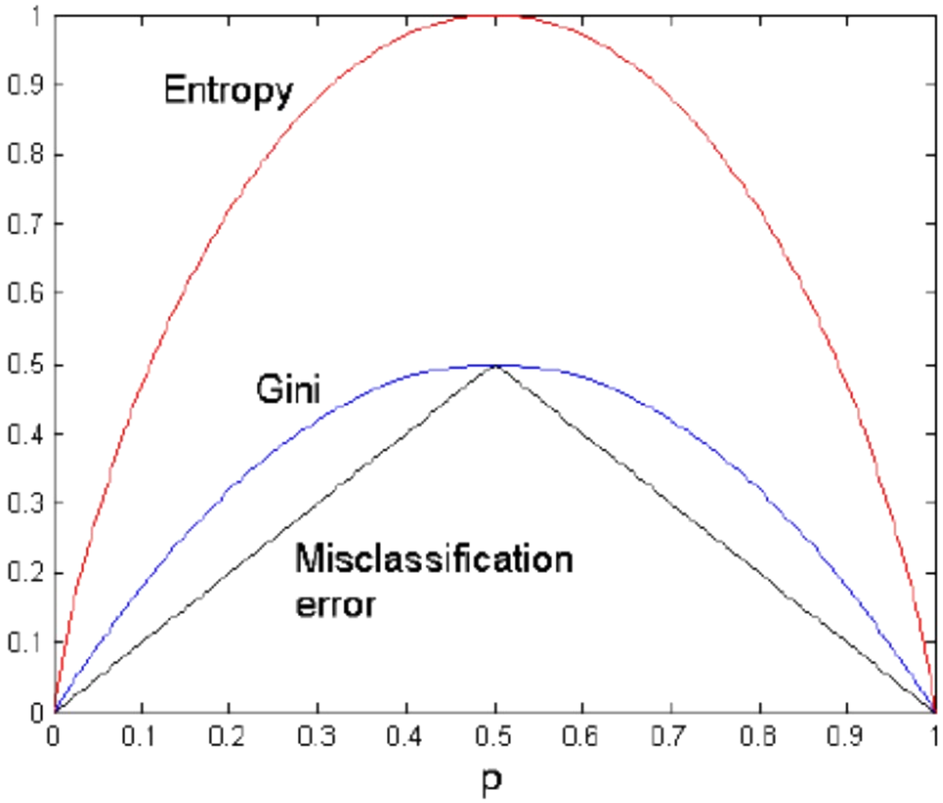
probability of being wrong when randomly estimating label by pulling one

$$G.I(A) = \sum_{i=1}^{d} \left[R_i \left(1 - \sum_{k=1}^{m} p_{ik}^2 \right) \right]$$

if the data in the set are all the same : Gini index = 0 (completely pure)

misclassification error

as shown on the right, there is non-differentiable point



Patgiri, Ripon & Nongmeikapam, Aditya. (2020). Empirical Study on Airline Delay Analysis and Prediction.

Model Learning

recursive partitioning

income	lot Size	ownership	income	lot size	ownership
60.0	18.4	Y	75.0	19.6	N
85.5	16.8	Y	52.8	20.8	N
64.8	21.6	Y	64.8	17.2	N
61.5	20.8	Y	43.2	20.4	N
87.0	23.6	Y	84.0	17.6	N
110.1	19.2	Y	49.2	17.6	N
108.0	17.6	Y	59.4	16.0	N
82.8	22.4	Y	66.0	18.4	N
69.0	20.0	Y	47.4	16.4	N
93.0	20.8	Y	33.0	18.8	N
51.0	22.0	Y	51.0	14.0	N
81.0	20.0	Y	63.0	14.8	N

whole dataset

entropy

$$-\frac{1}{2}\log_2\frac{1}{2}-\frac{1}{2}\log_2\frac{1}{2}=1$$

partitioning

51.0	14.0	N

and otherwise

entropy

$$\frac{1}{24}\left(-\log_2 1\right)+\frac{23}{24}\left(-\frac{12}{23}\log_2\frac{12}{23}-\frac{11}{23}\log_2\frac{11}{23}\right)\approx 0.96$$

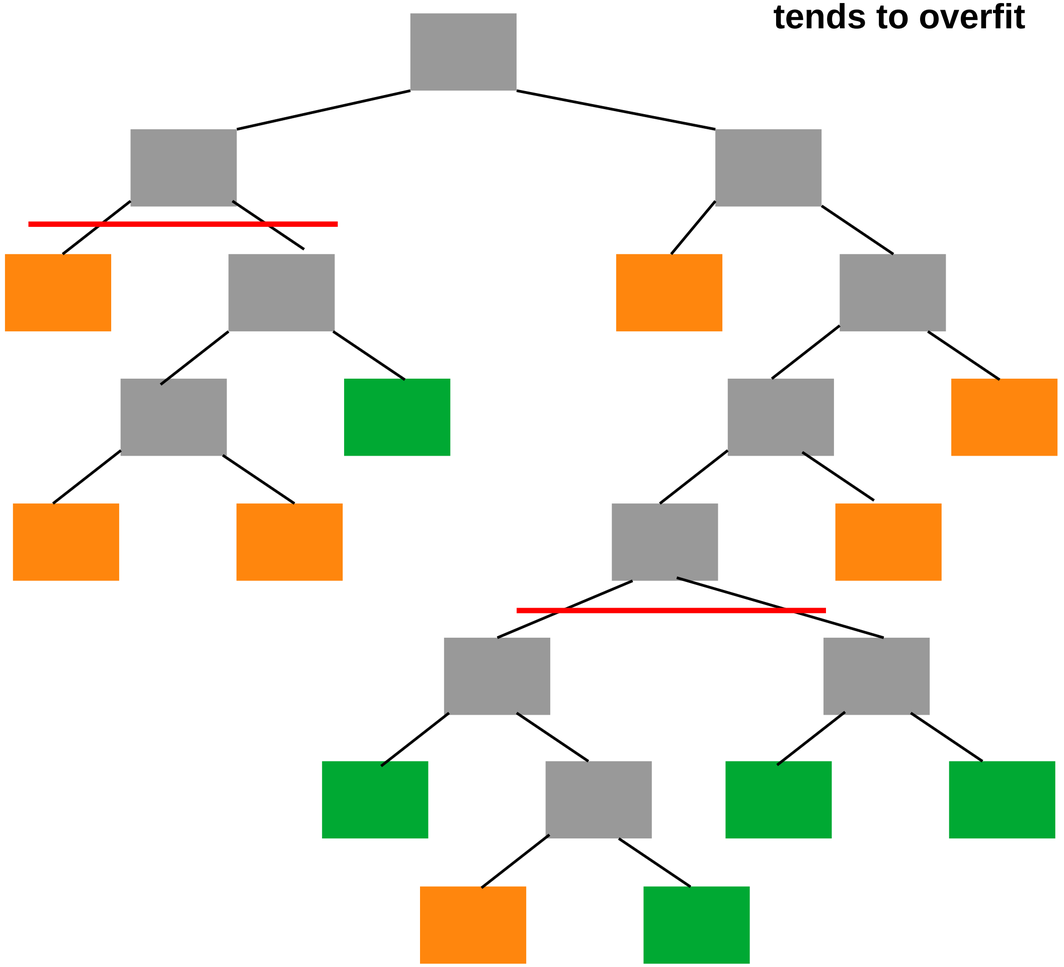
gained information : 0.04

Model Learning

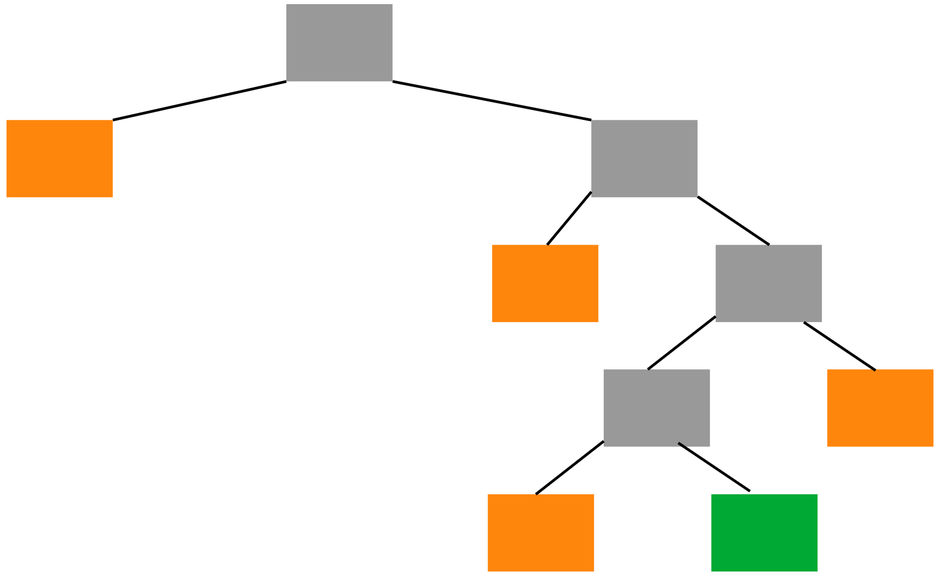
pruning

homogeneity of every terminal node is 1 : full tree

tends to overfit



pruned



cost function of pruning

$$CC(T) = Err(T) + \alpha \times L(T)$$

cost complexity

weight

number of terminal nodes

Further Reading

- high prediction performance, low complexity

- has persuasive power for each variable

but it’s only effective for certain data (decision boundary is parallel to data axis)

random forest

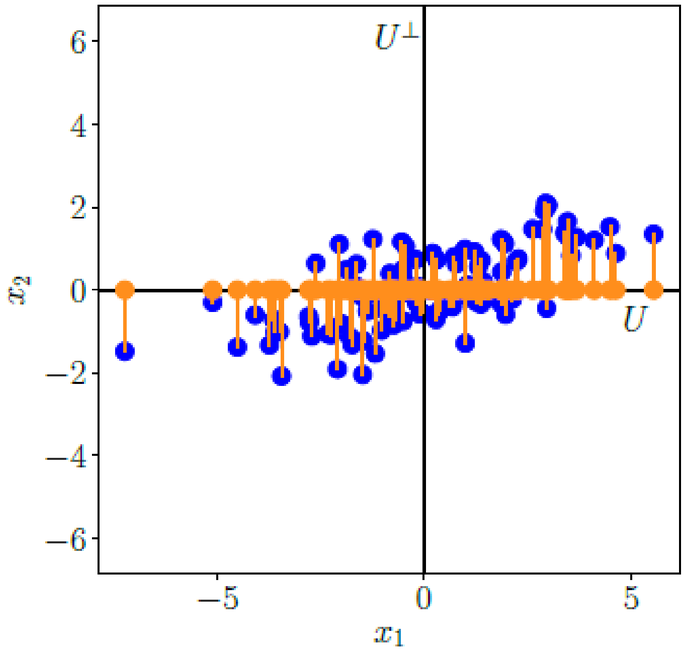
use many decision tree simultaneously

- sampling with replacement : make sub-datasets

- apply decision tree to each sub-dataset

⟶ increase diversity

rotation forest



rotate training data axis using PCA

preserve deviation, improve learning efficiency