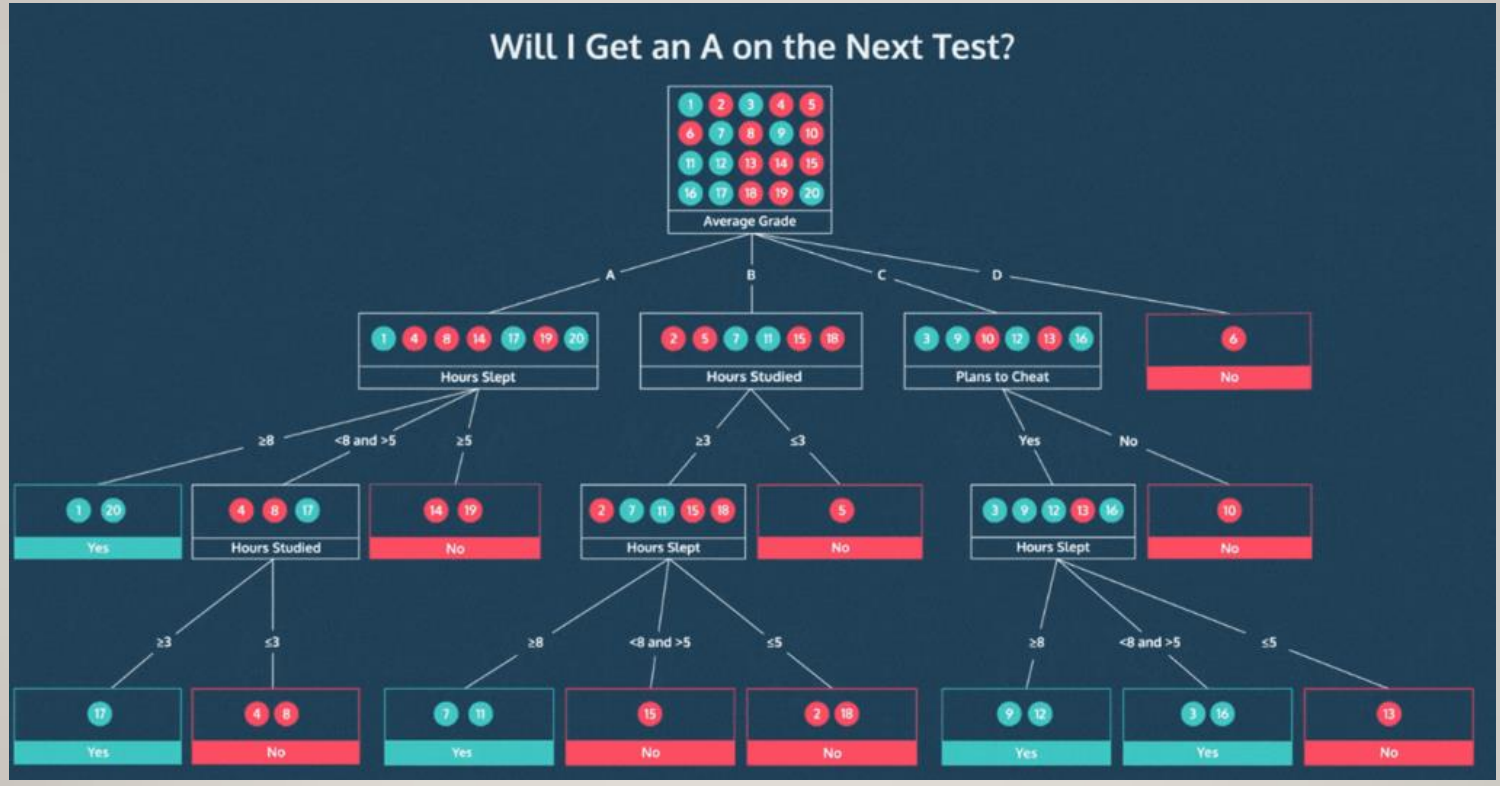
BSC LAB
Business & Service Computing

PNU Industrial Data Science

Decision Tree

Prof. Hyerim Bae

산업데이터 과학 성적



출처: https://hleecaster.com/ml-decision-tree-concept/

Contents

산업데이터과학은 산업현장에서 수집된 데이터를 분석하는데 필요한 기초 소양을 강의합니다.

Decision Tree

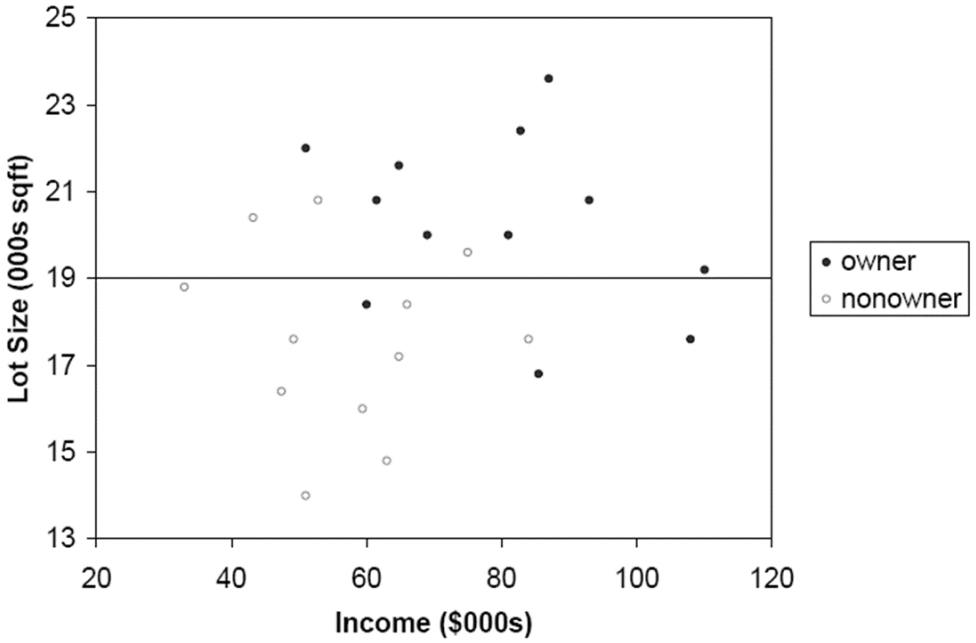
What is classification problem?

Goal: Classify or predict an outcome based on a set of predictors

The output is a set of **rules**

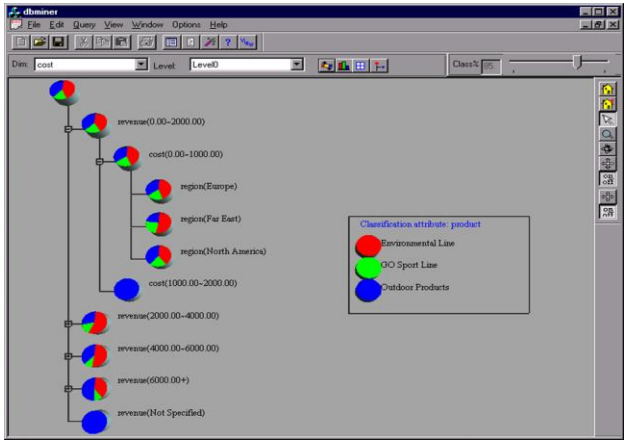
Example:

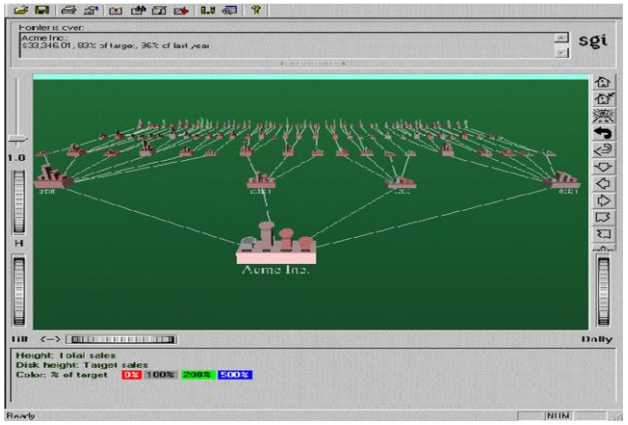
- Goal: classify a record as "will accept credit card offer" or "will not accept"
  - Rule might be "IF (Income > 92.5) AND (Education < 1.5) AND (Family <= 2.5) THEN Class = 0 (nonacceptor)



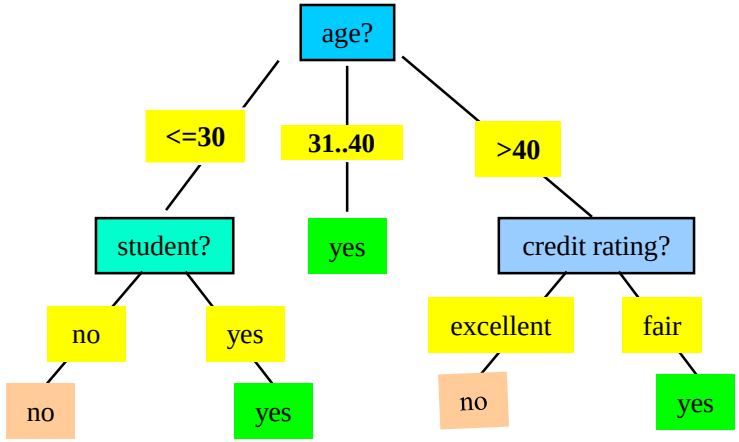
An example

- Hierarchical classification
  - Recursive partitioning
  - Pruning the tree





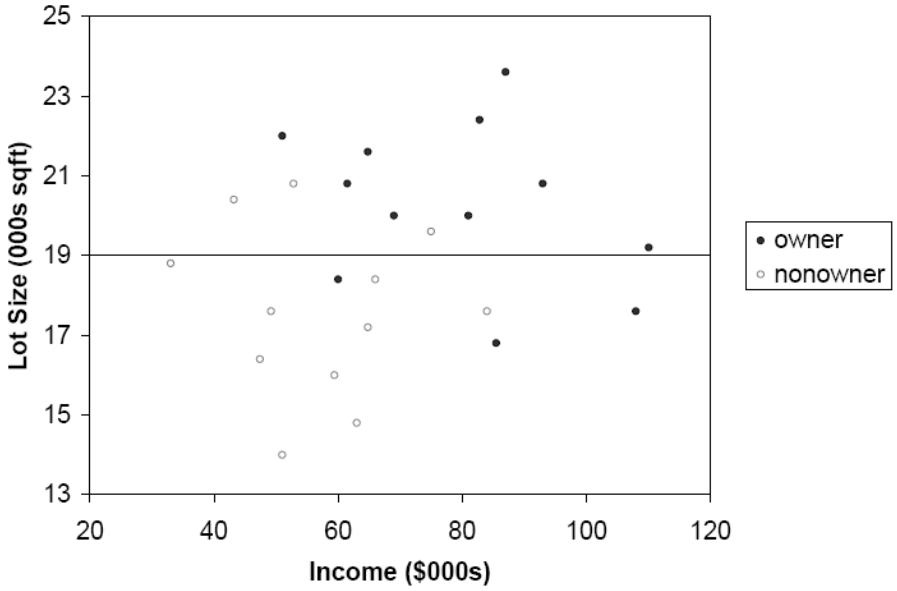
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

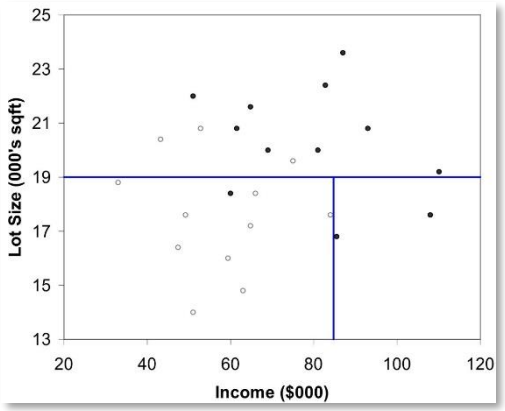


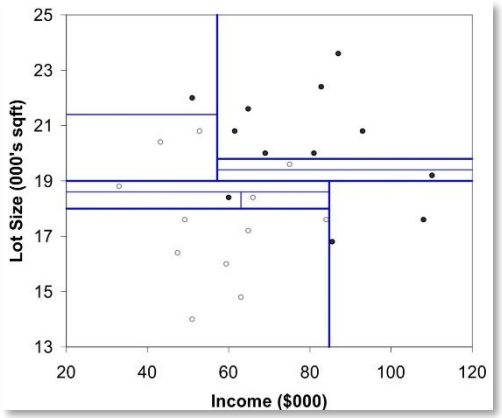
Recursive partitioning

- Pick one of the predictor variables, x_i
- Pick a value of x_i, say s_i, that divides the training data into two (not necessarily equal) portions
- Measure how "pure" or homogeneous each of the resulting portions are
  - "Pure" = containing records of mostly one class
- Algorithm tries different values of x_i and s_i to maximize purity in initial split
- After you get a "maximum purity" split, repeat the process for a second split, and so on
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

The first split



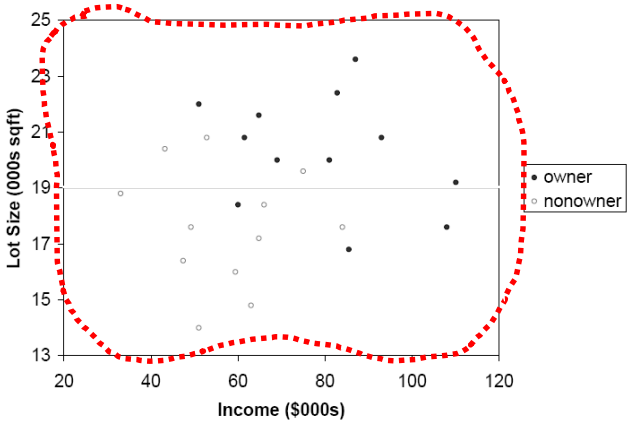




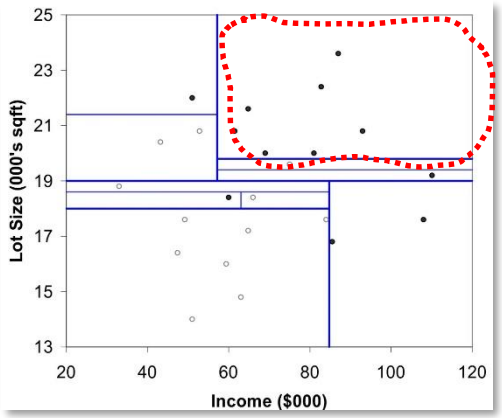
Measuring impurity

- How pure is the dataset?

Before split



After split



Gini Index

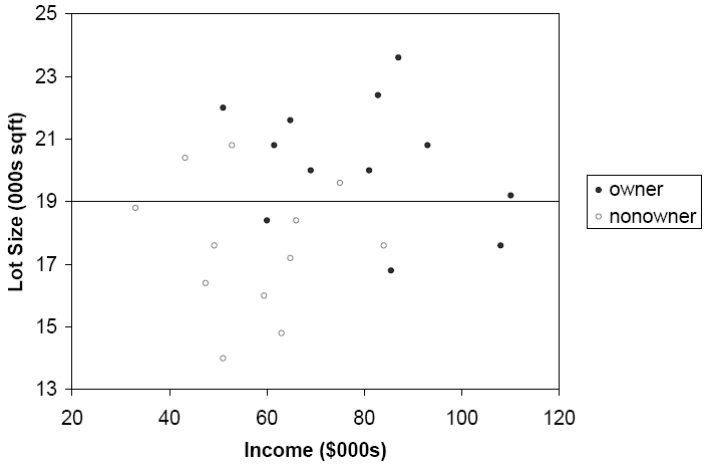
p = proportion of cases in rectangle A that belong to class k

Gini Index for rectangle A containing m records

$$I(A) = 1 - \sum_{k=1}^{m} p_k^2$$

- I(A) = 0 when all cases belong to same class
- Max value when all classes are equally represented (= 0.50 in binary case)

Note: XLMiner uses a variant called "delta splitting rule"

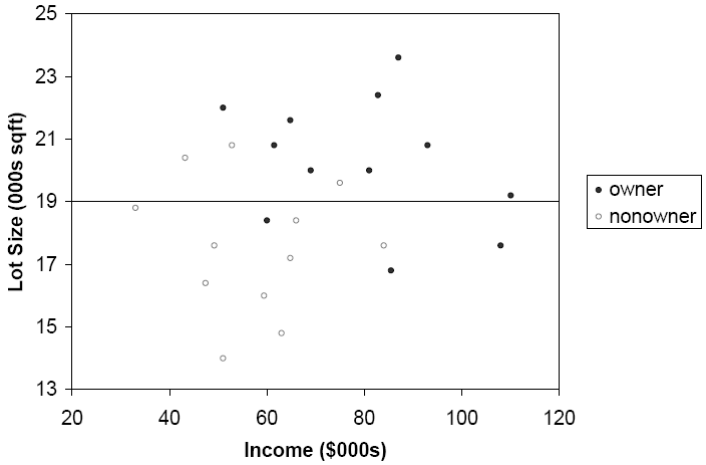


Entropy

p = proportion of cases (out of m) in rectangle A that belong to class k

- Entropy ranges between 0 (most pure) and $\log_2(m)$ (equal representation of classes)

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



Tree generation from data

	Height	Hair	Eyes	Class
1	short	blond	blue	+
2	tall	blond	brown	-
3	tall	red	blue	+
4	short	dark	blue	-
5	tall	dark	blue	-
6	tall	blond	blue	+
7	tall	dark	brown	-
8	short	blond	brown	-

- Problem
  - Given:
    - A set of training examples, all at once, each example characterized by a set of properties and a class to which the example belongs

    e.g.: height = {short, tall}

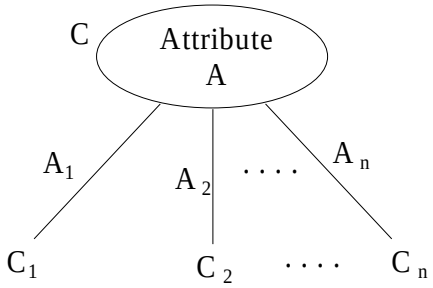
    hair = {dark, red, blond}

    eye = {brown, blue}

    class = {+, -}
  - Find:
    - A description tree (nodes labeled by attributes, arcs labeled by values, leaves labeled by a single class) of minimum size that correctly classifies the examples.

- Next, define

  $B(C, A) = [P\{\text{value}(A) = A_i\} * M(C_c)]$



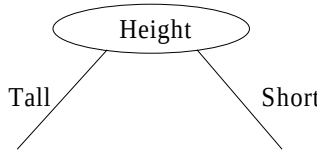
  - $B(C, A)$ is a measure of the new expected information content, i.e., the information we still need after classify C by attribute A.
- The attribute to test next is that which gains the most information, i.e., choose A to

  MAXIMIZE $M(C) - B(C, A)$

- *C* contains 3 elements of class + and 5 elements of class -

  So, $M(C) = -(3/8)\log(3/8) - (5/8)\log(5/8) = .954$

  Let's try testing the first attribute, 'height':



	tall blond brown -	short blond blue +	
	tall red blue +	short dark blue -	
i = tall	tall dark blue -	short blond brown -	*j* = short
	tall blond blue +		
	tall dark brown -		

$M(C_i) = -(2/5)\log(2/5) - (3/5)\log(3/5) = .971$

$M(C_j) = -(1/3)\log(1/3) - (2/3)\log(2/3) = .918$

$B(C, \text{'height'}) = (5/8)*.971 + (3/8)*.918 = .951$

$M(C) - B(C, \text{'height'}) = .954 - .951 = .003$

- Next, testing 'hair' :

  $B(C, \text{'hair'}) = (3/8)*0 + (1/8)*0 + (4/8)*1 = .5$

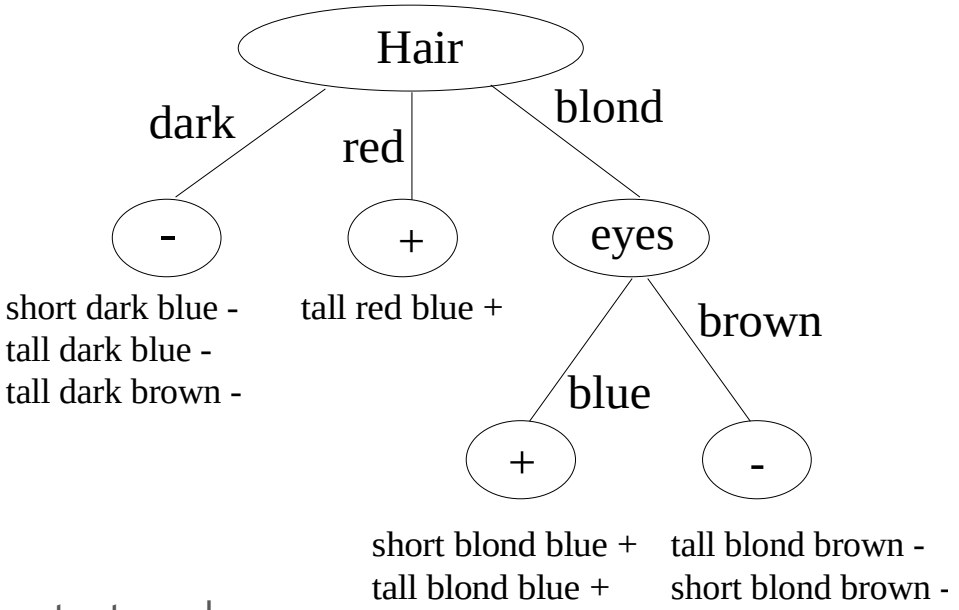
  $M(C) - B(C, \text{'hair'}) = .954 - .5 = .454$

  Finally, testing 'eyes' :

  $M(C) - B(C, \text{'eyes'}) = .347$

  - So the attribute to test first is HAIR.
  - Two of the three successors of 'hair' need no further information. The third successor (value 'blond') needs to be further divided, so we test 'height' and 'eyes' from that node, etc.

- A minimal decision tree



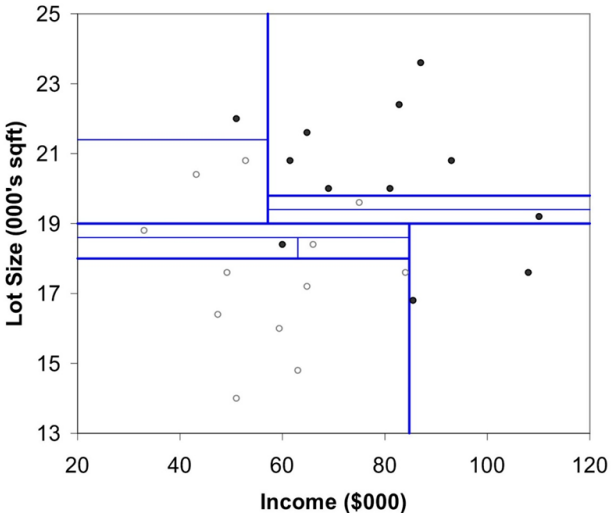
  - This tree is minimal, having only two test nodes.
  - Notice, also, that attribute 'height' is never tested at all.

Determining leaf node label

- Each leaf node label is determined by "voting" of the records within it, and by the cutoff value
- Records within each leaf node are from the training data
- Default cutoff=0.5 means that the leaf node's label is the majority class.
- Cutoff = 0.75: requires majority of 75% or more "1" records in the leaf to label it a "1" node

Overfitting tree

- Is the following model a good classifier?
  - In terms of purity
  - For training data
  - How about new data set?



Solving the problem

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: *Halt tree construction early* ‒ do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: *Remove branches* from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

CHAID

- Early termination
- CHAID, older than CART, uses chi-square statistical test to limit tree growth
- Splitting stops when purity improvement is not statistically significant

각 노드에서 반응변수와 가장 큰 연관성을 보이는 예측변수로 분할하는데, 이 연관성의 강도를 chi-square검증으로 찾아내고, 가장 연관성이 큰 예측변수가 통계적으로 유의한 개선결과를 보이지 못하면 중단

Pruning

- CART lets tree grow to full extent, then prunes it back
  - Idea is to find that point at which the validation error begins to rise
  - Generate successively smaller trees by pruning leaves
  - At each pruning stage, multiple trees are possible
- Use *cost complexity* to choose the best tree at that stage

  CC(T) = cost complexity of a tree
  Err(T) = proportion of misclassified records
  L(T) = the number of leaf node
  a = penalty factor attached to tree size (set by user)

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

  - Among trees of given size, choose the one with lowest CC
  - Do this for each size of tree

- Pruning process yields a set of trees of different sizes and associated error rates

  Two trees of interest:
  - Minimum error tree
    Has lowest error rate on validation data
  - Best pruned tree
    Smallest tree within one std. error of min. error
    This adds a bonus for simplicity/parsimony

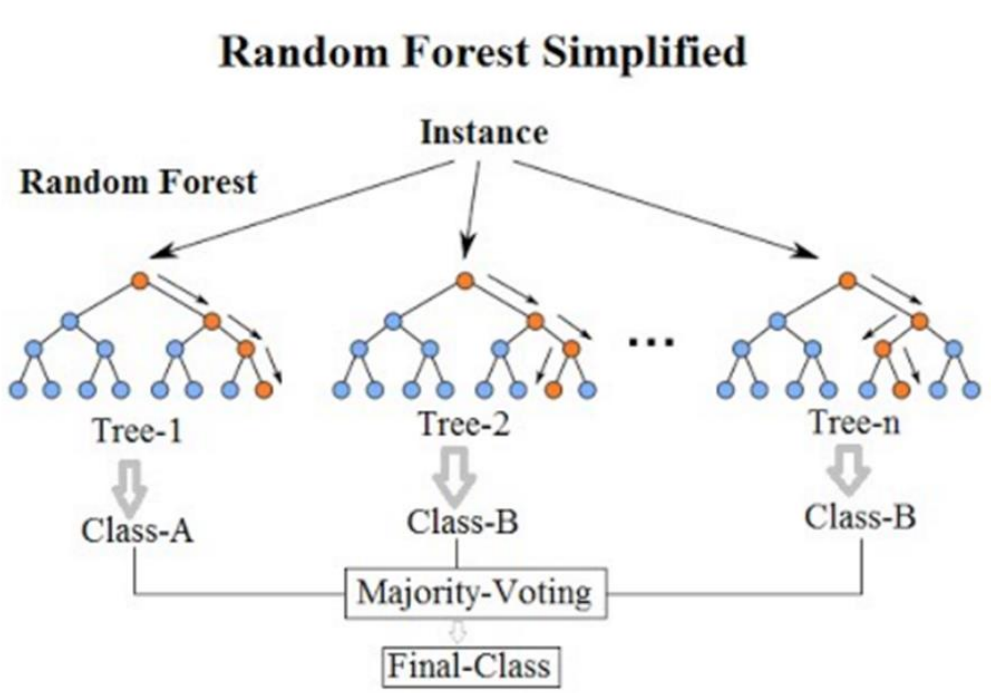
Regression tree for prediction

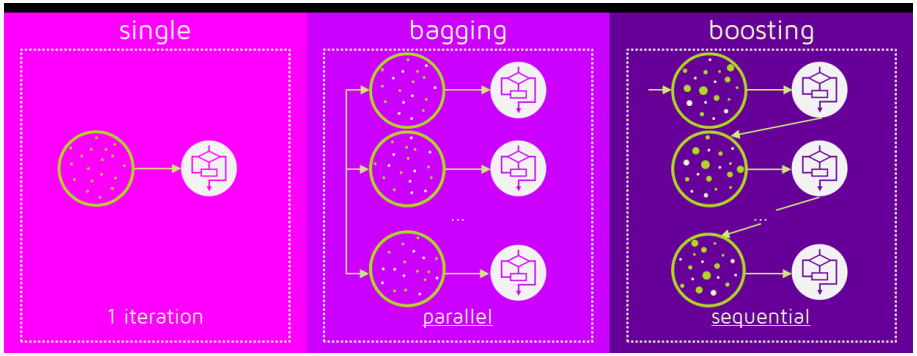
- Used with continuous outcome variable
- Procedure similar to classification tree
- Many splits attempted, choose the one that minimizes impurity

- Prediction is computed as the **average** of numerical target variable in the rectangle (in CT it is majority vote)
- Impurity measured by **sum of squared deviations** from leaf mean
- Performance measured by RMSE (root mean squared error)

Random Forest

- Ensemble
  - Bagging
  - Boosting





Source: swallow.github.io

Advantages of trees

- Easy to use, understand
- Produce rules that are easy to interpret & implement
- Variable selection & reduction is automatic
- Do not require the assumptions of statistical models
- Can work without extensive handling of missing data

Disadvantages

- May not perform well where there is structure in the data that is not well captured by horizontal or vertical splits
- Since the process deals with one variable at a time, no way to capture interactions between variables

Summary

- Classification and Regression Trees are an easily understandable and transparent method for predicting or classifying new records
- A tree is a graphical representation of a set of rules
- Trees must be pruned to avoid over-fitting of the training data
- As trees do not make any assumptions about the data structure, they usually require large samples