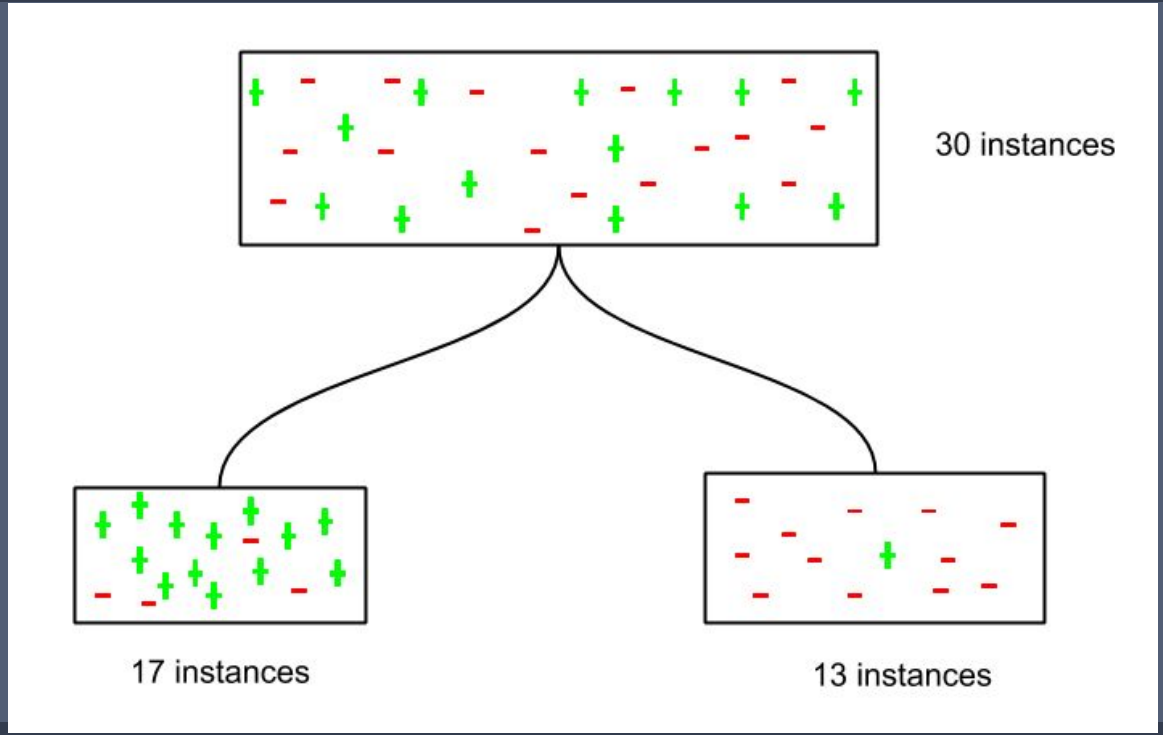
Index Class

- Regression trees
- Other trees
- Classification trees
- Bagged trees
- Random Forest

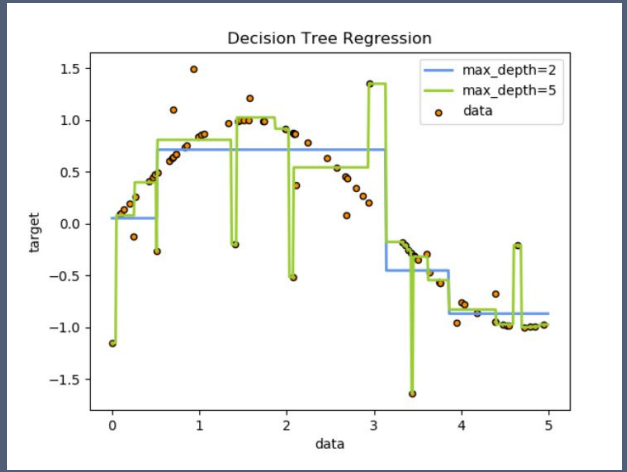
Trees



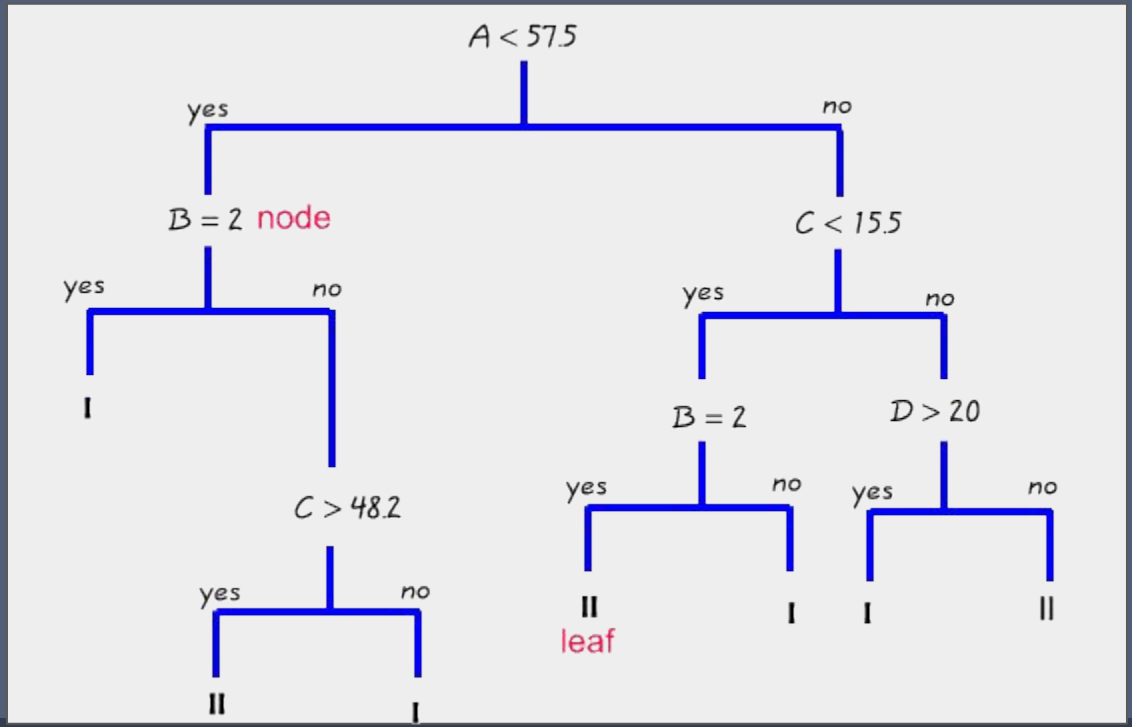
Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.

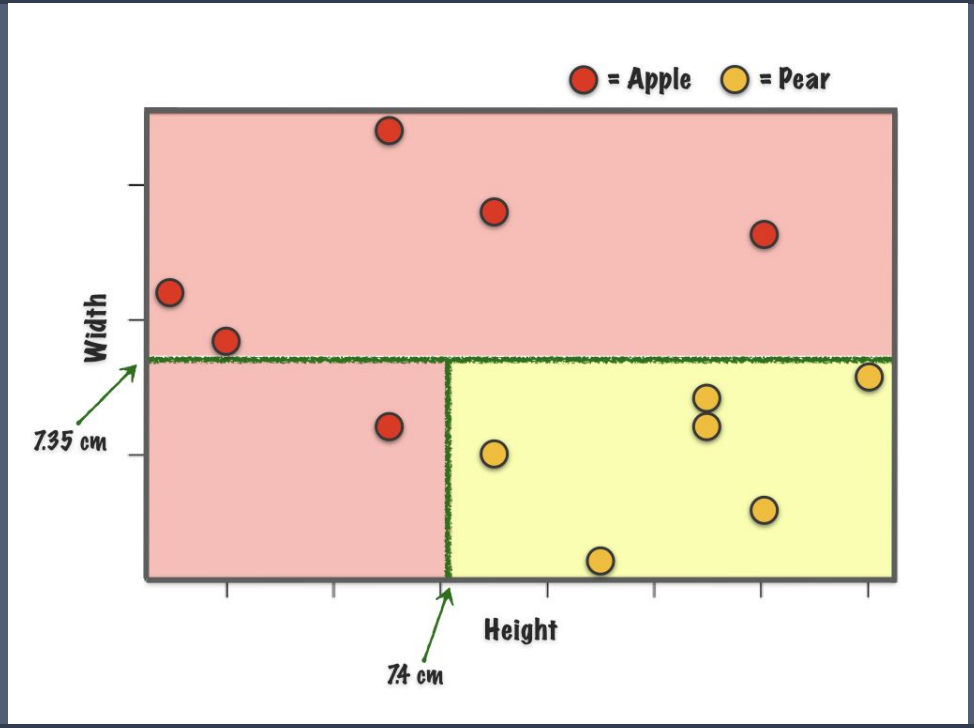
A tree has splits, nodes and leafs

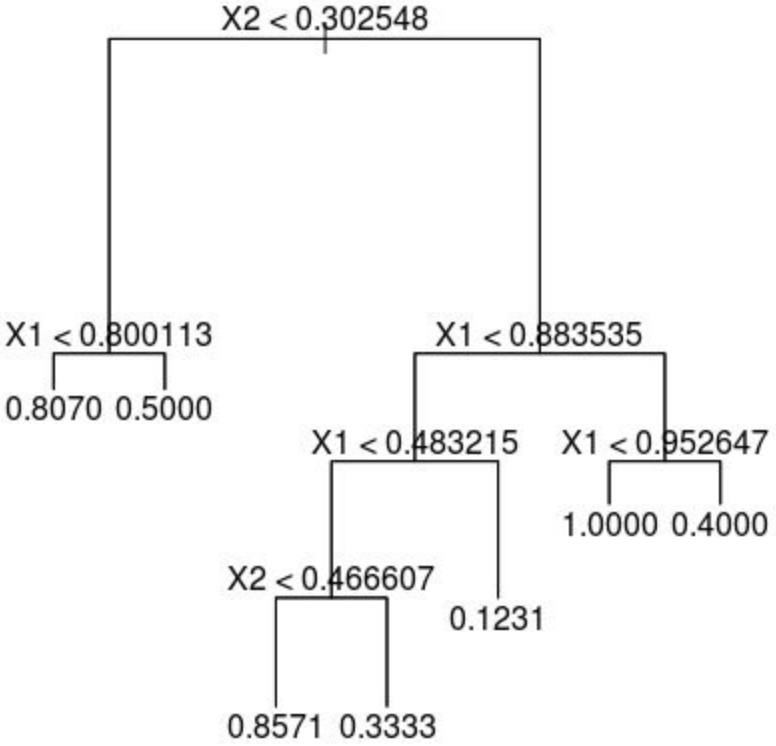


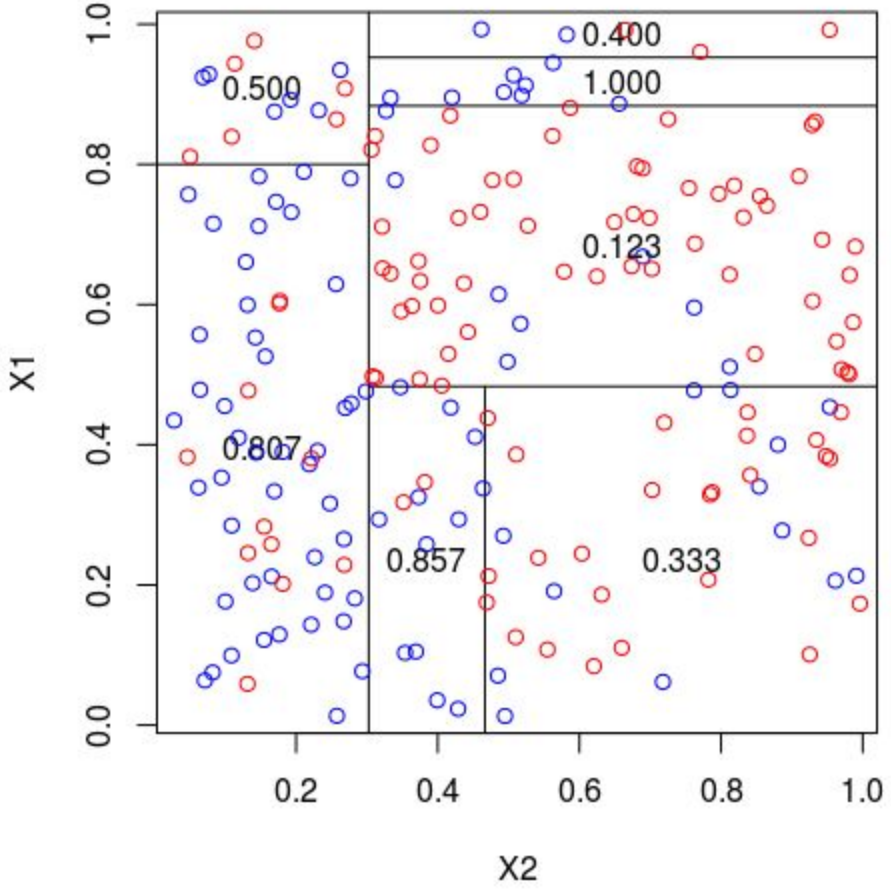
Trees


A < 57.5
yes
no
B = 2 node
C < 15.5
yes
no
yes
no
I
B = 2
D > 20
C > 48.2
yes
no
yes
no
yes
no
II
I
II
leaf
I
II

Trees




X2 < 0.302548
X1 < 0.800113
0.8070 0.5000
X1 < 0.883535
X1 < 0.483215
X1 < 0.952647
1.0000 0.4000
X2 < 0.466607
0.1231
0.8571 0.3333


0.500
0.400
1.000
0.123
0.807
0.857
0.333
X1
X2
0.0
0.2
0.4
0.6
0.8
1.0

Pros

- Decision trees are easy to interpret and visualize.
- It can easily capture Non-linear patterns.
- It requires fewer data preprocessing from the user
- The decision tree has no assumptions about distribution
- Allows missing data (surrogate splits)

Cons

- Sensitive to noisy data. It can overfit noisy data
- The small variation (or variance) in data can result in a different decision tree
- Interpretation problems with correlated data
- They are not good if the relation predictors-target is not in rectangular form

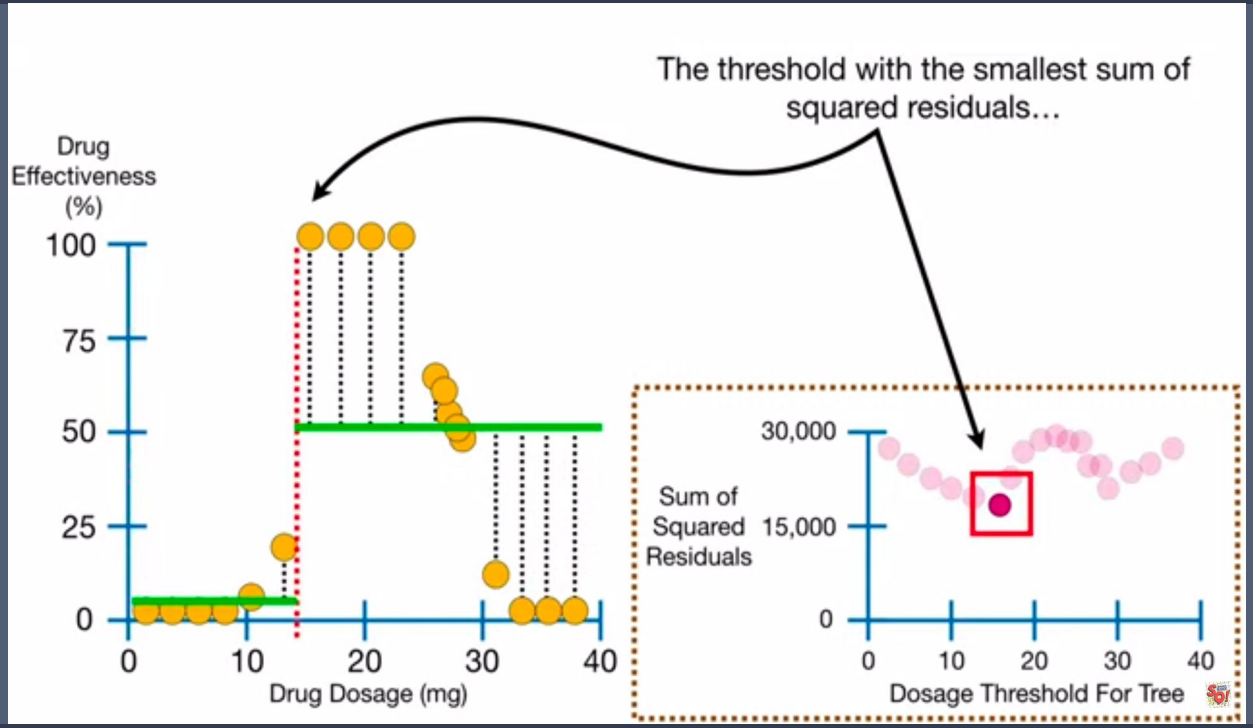
CART - A new split

CART Classification and regression trees

- The algorithm begins with all the dataset
- Search every distinct value of each predictor to divide the dataset in two sets
- Finish when some threshold is achieved

$$\text{SSE} = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2,$$

Regression trees - New Split



CART - Threshold

- Maximum depth of the tree
- Minimum n° samples for split
- Minimum n° samples for each leaf
- Maximum number of leafs
- Impurity : Impurity is a measure of how badly the observations at a given node fit the model

Complexity parameter - Prune

Once the tree is grown, the tree may overfit the data.
The cost of the tree may be reformulated to prune the tree using a cost-complexity parameter

$$\mathrm{SSE}_{c_p} = \mathrm{SSE} + c_p \times (\#\ \text{Terminal Nodes}),$$

Classification trees - impurity

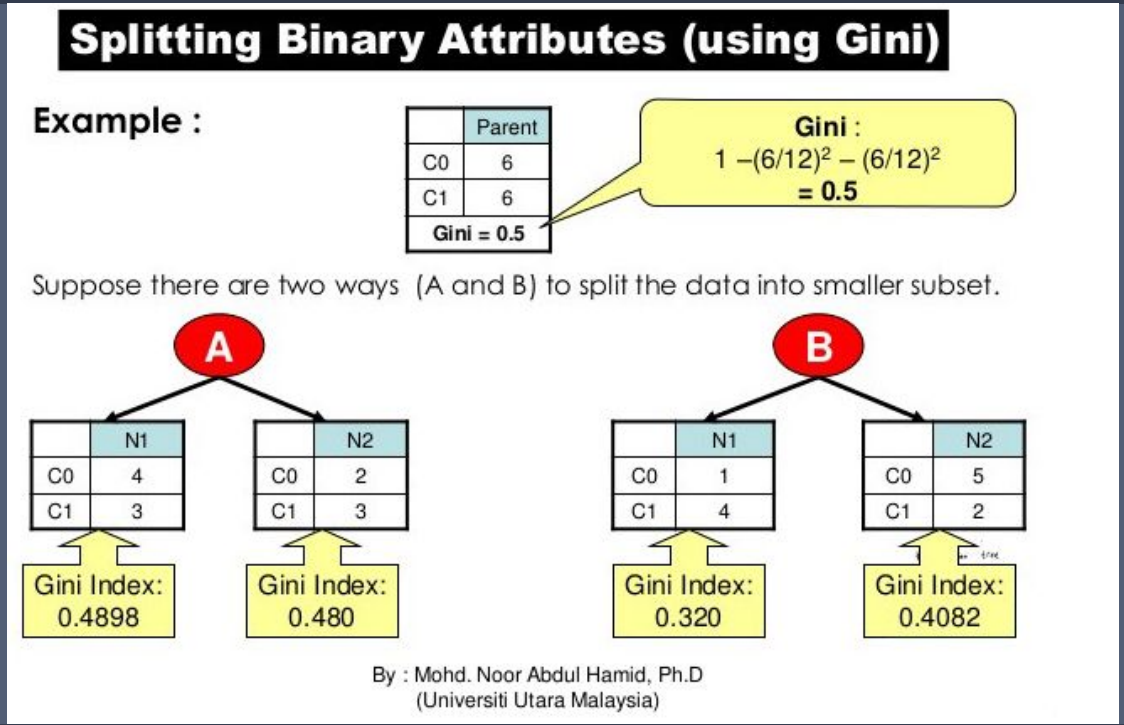
- Gini index

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

- Cross Entropy

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Classification trees - purity



Random Forest

1 Select the number of models to build, m
2 **for** $i = 1$ *to* m **do**
3 Generate a bootstrap sample of the original data
4 Train a tree model on this sample
5 **for** *each split* **do**
6 Randomly select k ($< P$) of the original predictors
7 Select the best predictor among the k predictors and partition the data
8 **end**
9 Use typical tree model stopping criteria to determine when a tree is complete (but do not prune)
10 **end**

Algorithm 8.2: Basic Random Forests

Random Forest

- A good number of trees is 1000
- The predictors are chosen randomly
- The number of predictors chosen is normally ⅓ of the total number
- They are robust to outliers and noise
- Train a random forest is usually faster than bagged trees (Because the numbers of predictors is reduced)