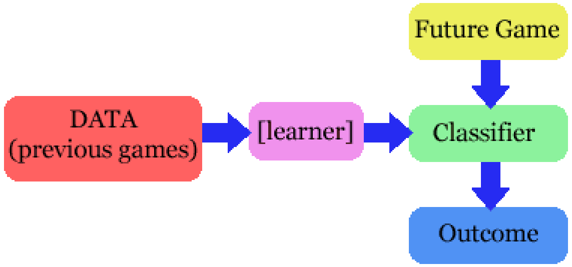
Decision Trees

Jiahui Chen
Department of Mathematical Sciences
University of Arkansas

Introduction

- Decision tree is a basic machine learning method
- Given training set to set up a model and then
  - Classification
  - Regression

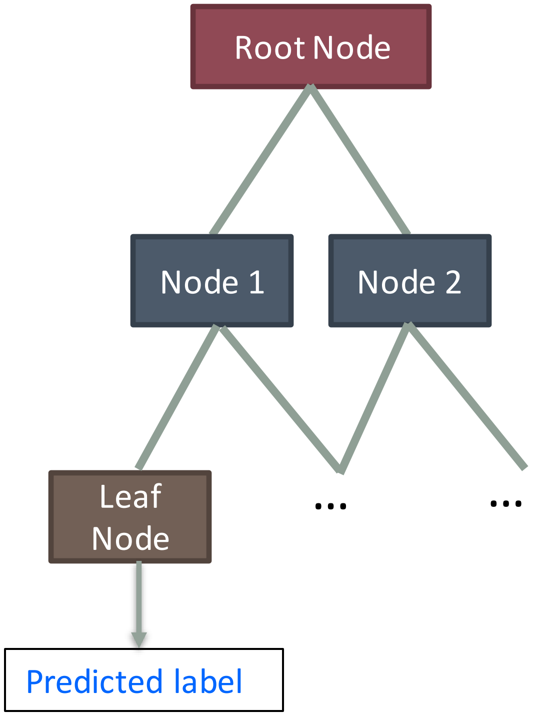


Introduction

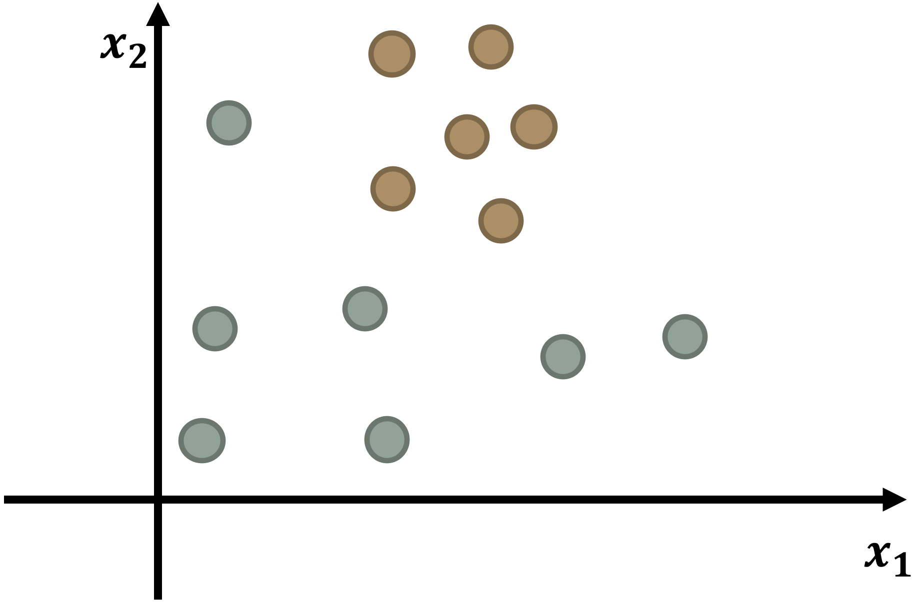
- Decision tree represents the attribute of the data using a flowchart-like structure
- Decision trees include

Root node, nodes (non-leaf nodes), and leaf nodes.

- Each node represents a condition to split the data
- In leaf node, we stop splitting the data and choose a label to represent all the data in the Leaf node

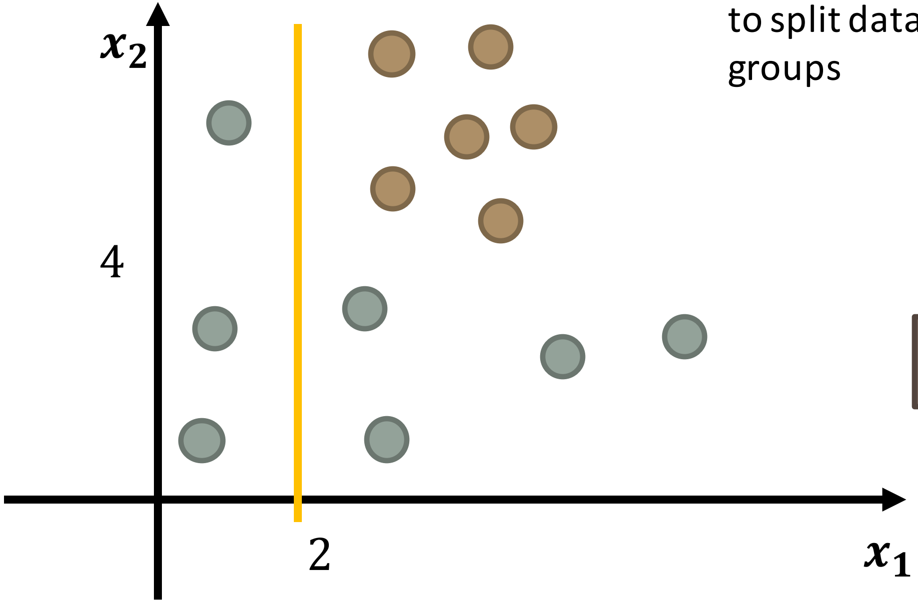


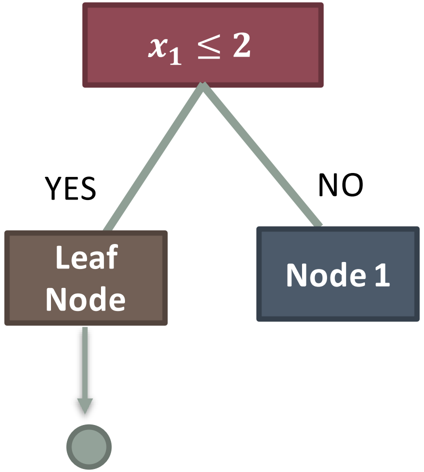
Example



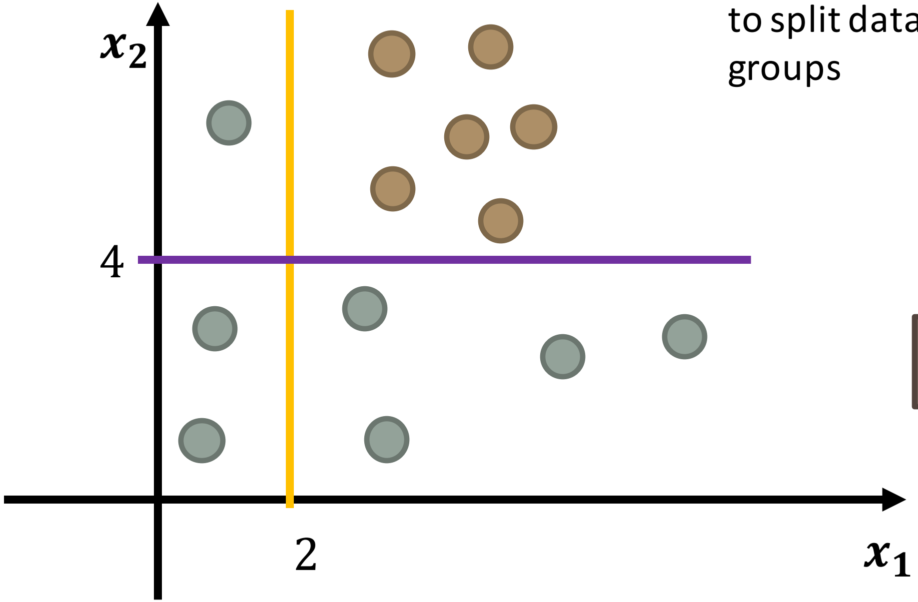
Example

Would like to construct a decision tree to split data into blue and red-circle groups

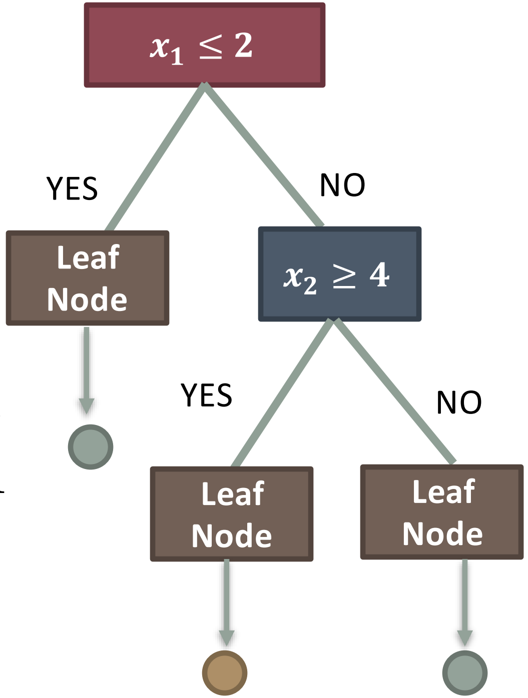




Example

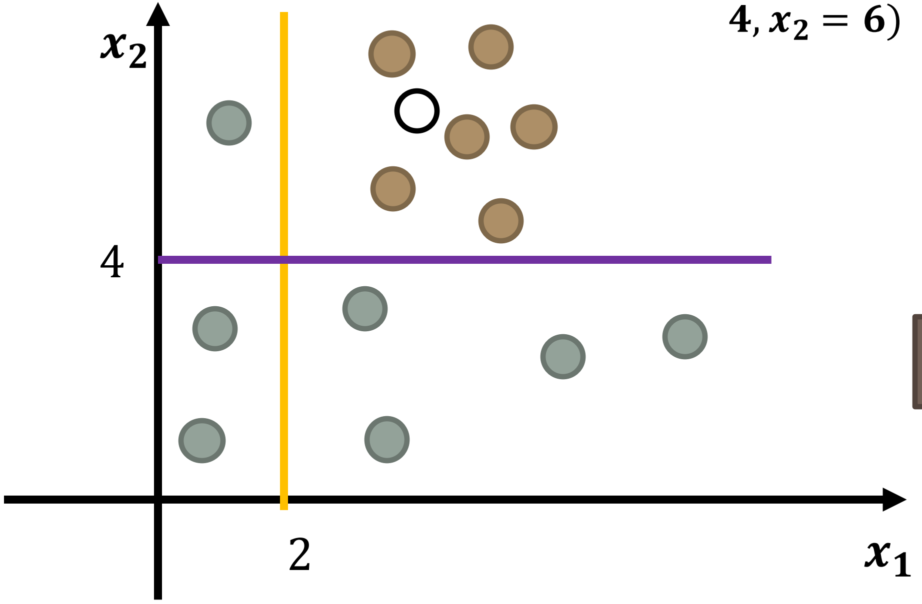


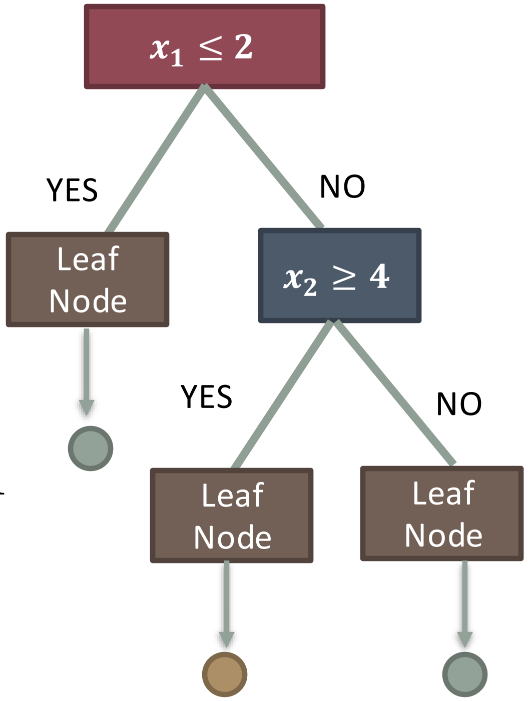
Would like to construct a decision tree to split data into blue and red-circle groups



Example -- A Test

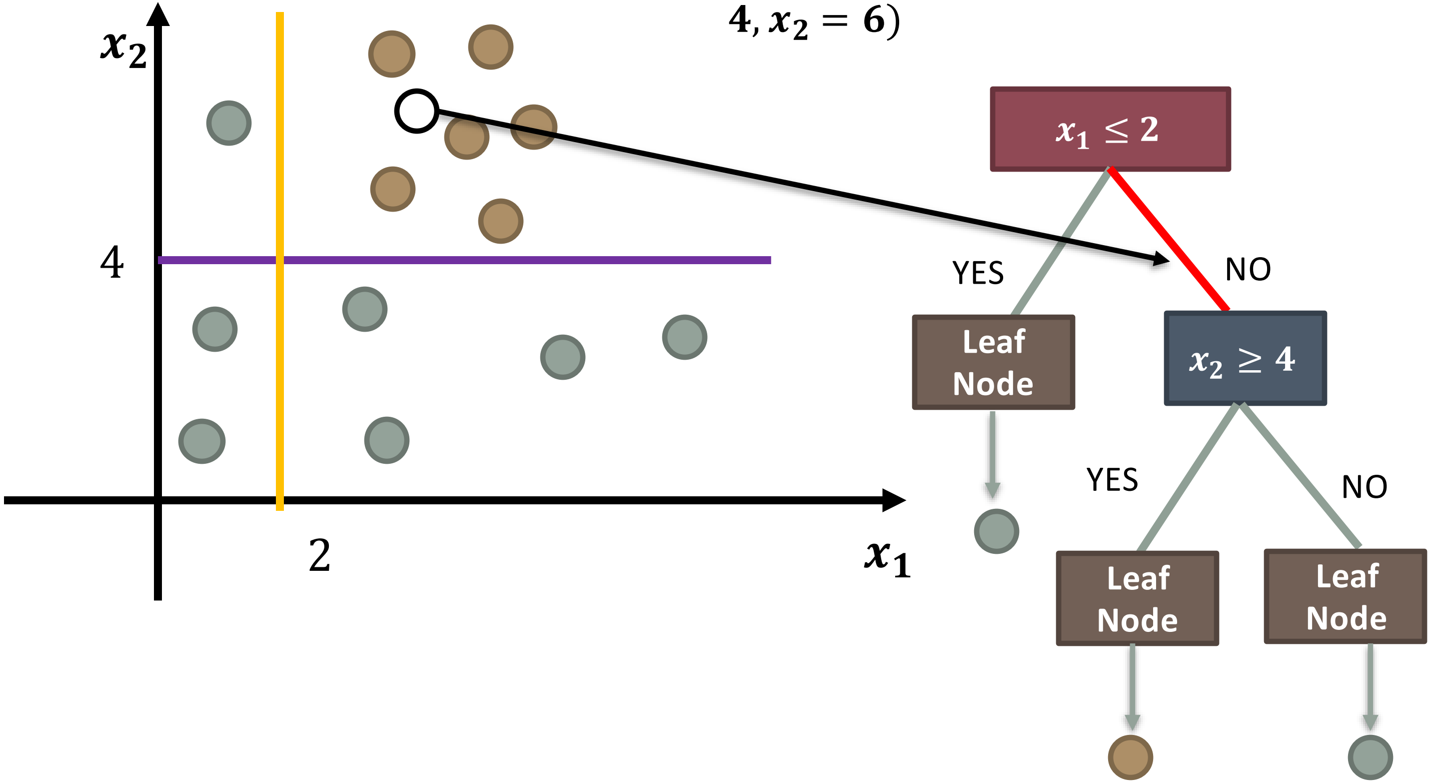
Assume we have a test data ($\boldsymbol{x_1} = \boldsymbol{4}, \boldsymbol{x_2} = \boldsymbol{6}$)





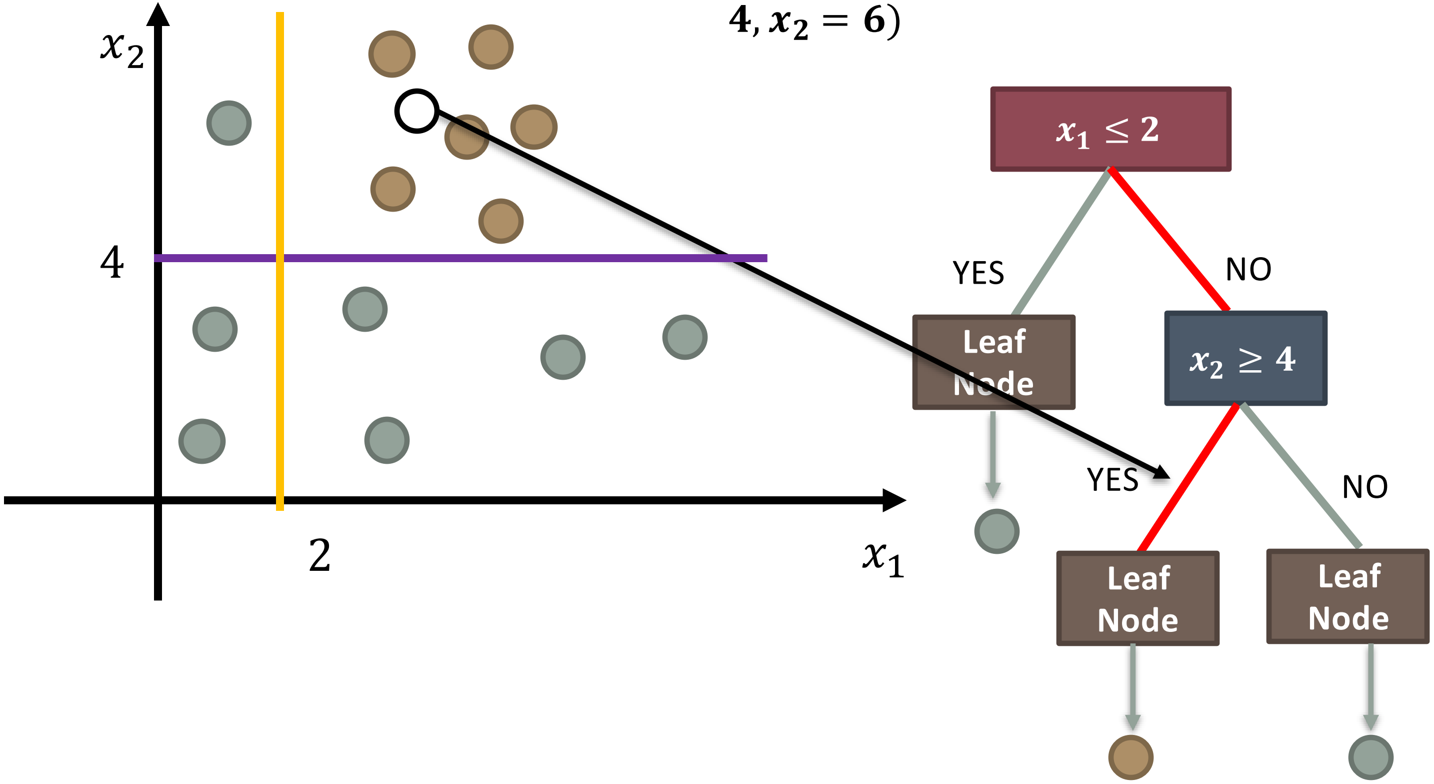
Example

Assume we have a test data ($\boldsymbol{x_1 = 4}, \boldsymbol{x_2 = 6}$)



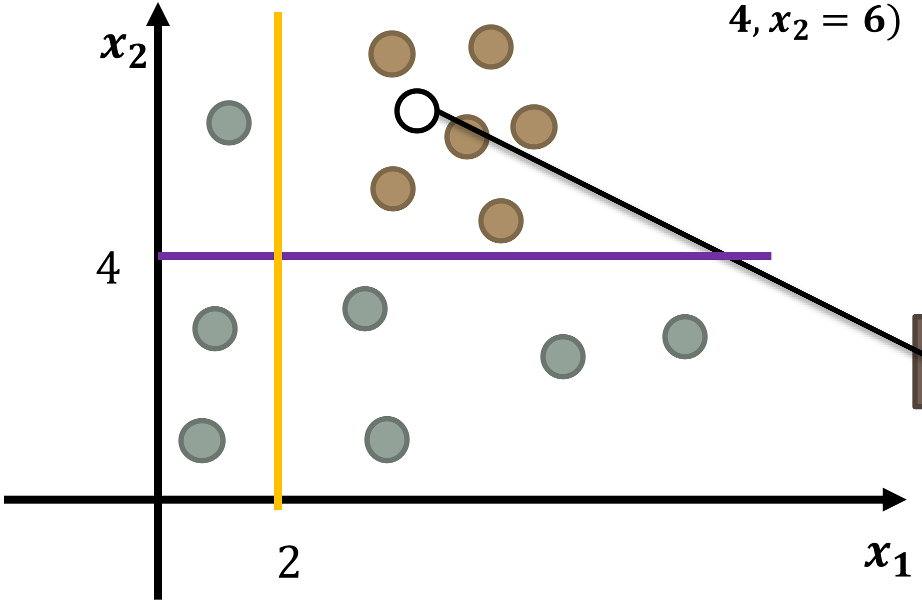
Example

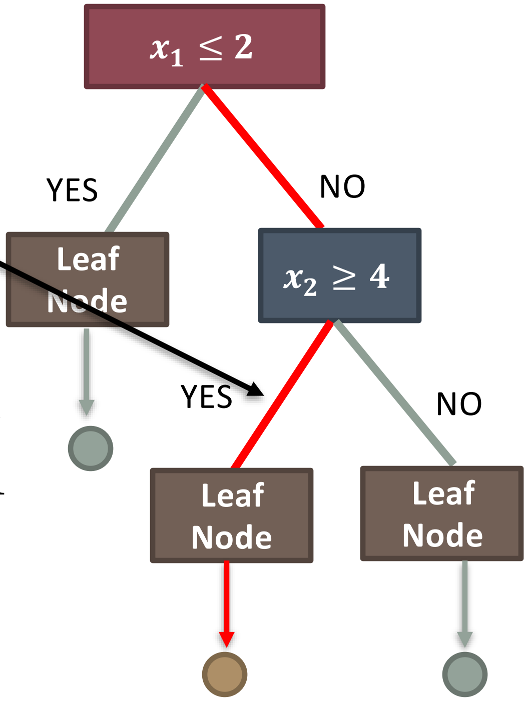
Assume we have a test data ($\boldsymbol{x_1} = \boldsymbol{4}, \boldsymbol{x_2} = \boldsymbol{6}$)



Example

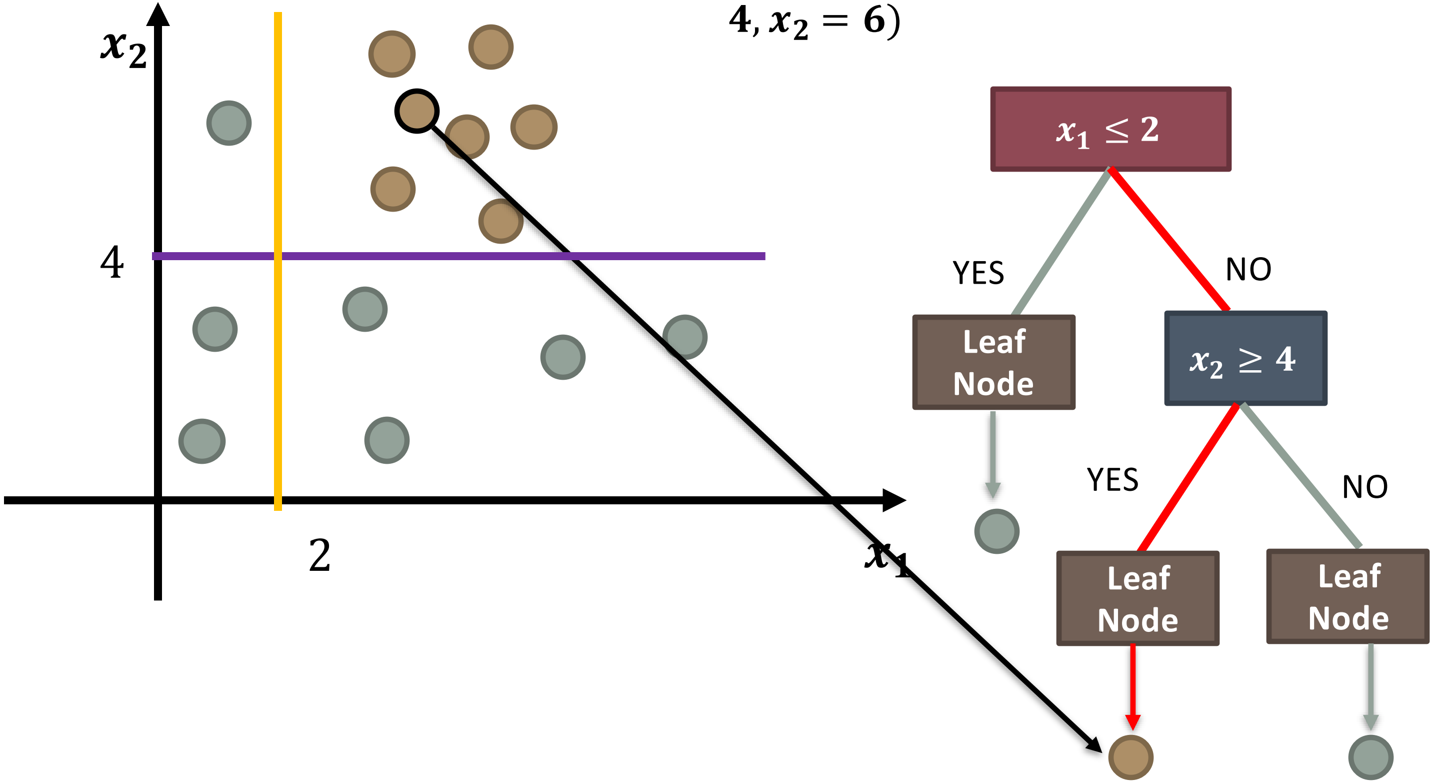
Assume we have a test data $(\boldsymbol{x_1} = \boldsymbol{4}, \boldsymbol{x_2} = \boldsymbol{6})$



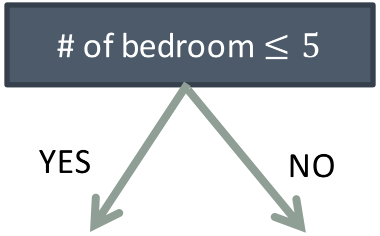


Example

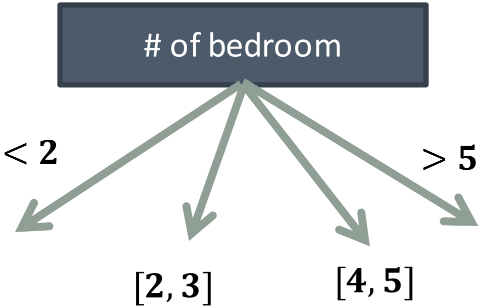
Assume we have a test data ($\boldsymbol{x_1} = \boldsymbol{4}, \boldsymbol{x_2} = \boldsymbol{6}$)



How to Split Data at Each Node



Binary split



Multi-way split

Tree Induction

- **Hunt's algorithm** (earliest one)
- **CART** (Classification And Regression Tree)
- **ID3, C4.5, C5.0** (use information gain)
- **CHAID** (CHi-squared Automatic Interaction Detection)
- **MARS** (Improvement for numerical features)
- **SLIQ, SPRINT**
- **Conditional Inference Trees** (recursive partition using statistical tests)

Slide adapted from Tan, Steinbach, Kumar

Impurity of a Node

Node 1:	Node 2:
Label 0: 5	Label 0: 9
Label 1: 5	Label 1: 1

Node 1 has a high degree of impurity

Node 2 has a low degree of impurity

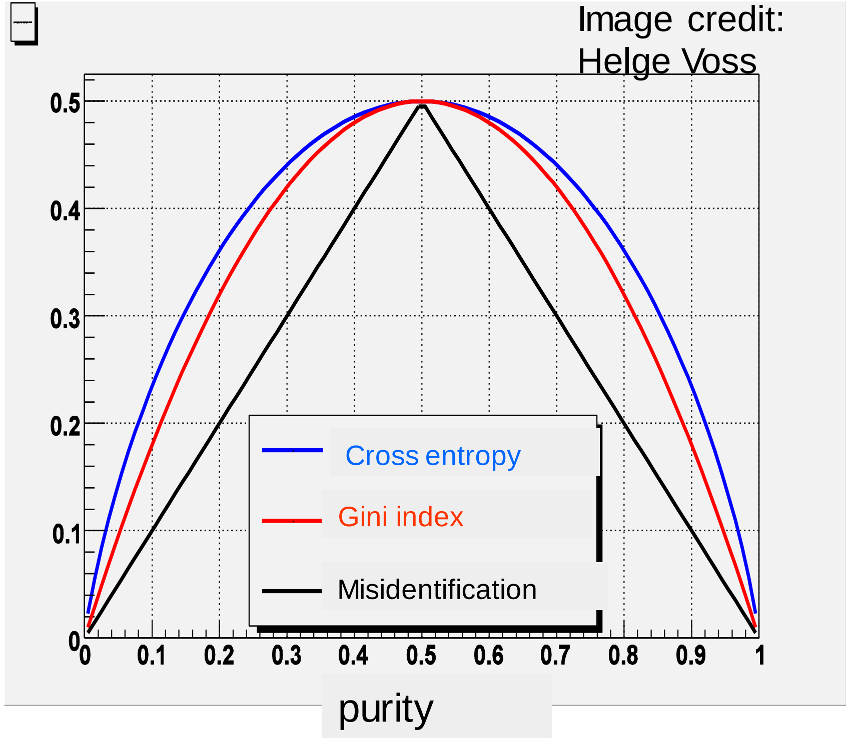
- We prefer a node with a low degree of impurity

How to Measure Node's Impurity

- **Classification**
  - Gini
  - Cross-entropy
  - Misclassification
- **Regression**
  - Mean squared error (standard deviation)
  - Mean absolute error

Decision Tree Classification

- **Classification**
  - **Gini index**
  - **Cross-entropy**
  - **Misclassification**



Gini index is the most used method!

Measure Node Impurity by GINI

- Gini index for a given node $\boldsymbol{t}$

$$\mathbf{GINI}(\boldsymbol{t}) = \sum_{j} \boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t})(\mathbf{1} - \boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t}))$$

$$= \mathbf{1} - \sum_{j} \boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t})^{\mathbf{2}}$$

Where $\boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t})$ is considered as the relative frequency of class $\boldsymbol{j}$ in node $\boldsymbol{t}$ (i.e., the probability of label j being chosen). Here $(\mathbf{1} - \boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t}))$ is probability that the choice is incorrect.

Measure Node Impurity by GINI

Node 1:
Label 0: 5
Label 1: 5

$$p(0|1) = \frac{5}{10} = 0.5$$

$$p(1|1) = \frac{5}{10} = 0.5$$

$$\mathrm{GINI}(1) = 1 - 0.5^2 - 0.5^2 = 0.5$$

Node 2:
Label 0: 9
Label 1: 1

$$p(0|1) = \frac{9}{10} = 0.9$$

$$p(1|1) = \frac{1}{10} = 0.1$$

$$\mathrm{GINI}(2) = 1 - 0.9^2 - 0.1^2 = 0.18$$

- We prefer a node with a lower GINI index

UNIVERSITY OF ARKANSAS

Example

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Cool	Normal	Weak	Yes

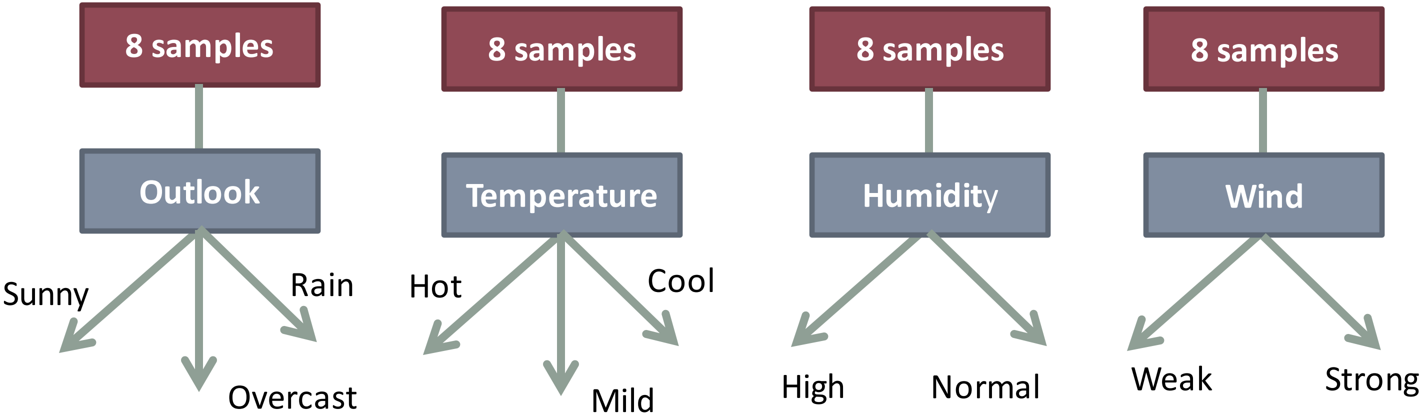
https://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm

Example

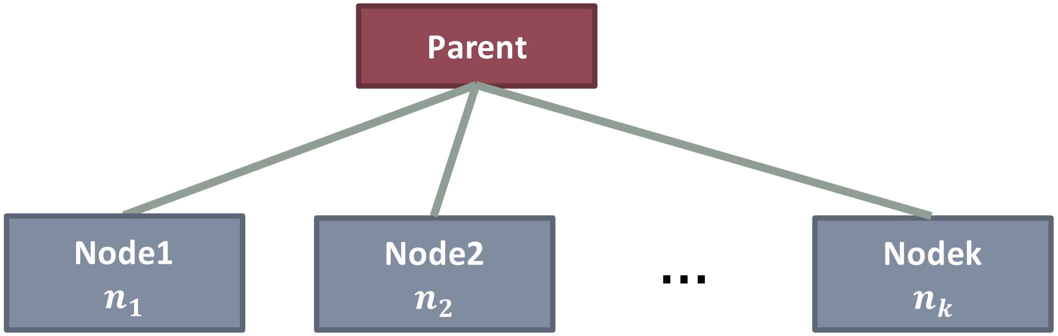
Summary:

- Outlook has 3 values: sunny, overcast, rain
- Temperature has 3 values: hot, mild, cool
- Humidity has 2 values: normal, high
- Wind has 2 values: weak, strong
- 2 Labels: No, Yes

Define the best split



Gain Defines Best Split



$$\text{Gain} = \text{Gini}(\text{Parent}) - \frac{n_1}{\sum n_i}\text{Gini}(\text{Node 1}) - \frac{n_2}{\sum n_i}\text{Gini}(\text{Node 2}) - \cdots - \frac{n_k}{\sum n_i}\text{Gini}(\text{Node k})$$

Measure Node Impurity by Entropy

- Entropy at a given node $\boldsymbol{t}$

$$\textbf{Entropy}(\boldsymbol{t}) = -\sum_{j} \boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t}) \log_2 \boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t})$$

Where $\boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t})$ is considered as the relative frequency of class $\boldsymbol{j}$ in node $\boldsymbol{t}$

- Entropy is originally is used to measure the uncertainty of a variable or information of a message
- $0 \log_2 0 = 0$
- The split the highest entropy will be taken at each step, until entropy is zero (i.e., children notes are pure).

Measure Node Impurity by Classification Error

- Classification error at a given node $\boldsymbol{t}$

$$\mathbf{Error}(\boldsymbol{t}) = \mathbf{1} - \max \boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t})$$

Where $\boldsymbol{p}(\boldsymbol{j}|\boldsymbol{t})$ is considered as the relative frequency of class $\boldsymbol{j}$ in node $\boldsymbol{t}$

Decision Tree Regression

Day	Outlook	Temperature	Humidity	Wind	Mins Played
D1	Sunny	Hot	High	Weak	25
D2	Sunny	Hot	High	Strong	30
D3	Overcast	Hot	High	Weak	48
D4	Rain	Mild	High	Weak	50
D5	Rain	Cool	Normal	Weak	60
D6	Rain	Cool	Normal	Strong	28
D7	Overcast	Cool	Normal	Strong	52
D8	Sunny	Cool	Normal	Weak	55

Use standard deviation to measure node impurity

When to Stop Splitting

- Stop splitting when all entries belong to the same class
- Stop splitting when all entries have the same features used for splitting conditions
- Stop splitting when the maximum tree depth has reached
- Stop splitting when the maximum number of nodes has reached.
- Termination criterion: Pre-Pruning and Post-Pruning

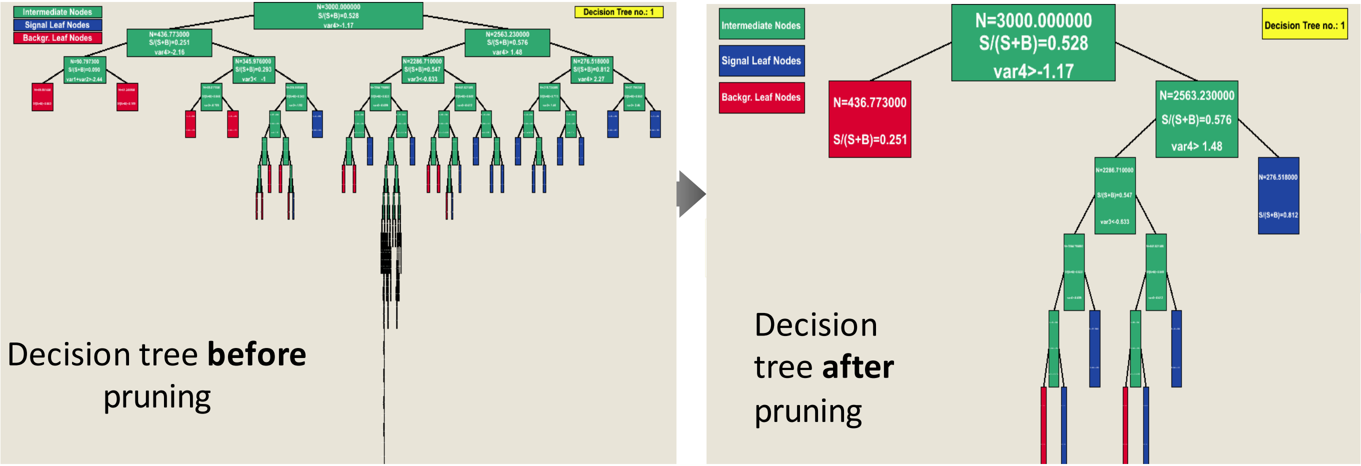
Pre-Pruning and Post-Pruning

- **Pre-Pruning**
  - Stop if number of entries in this node less than some user-specified threshold
  - Stop if class distribution **is independent of the available features** (use χ^2 test)
  - Stop if splitting **does not improve impurity measures**
- **Post-Pruning**
  1. Grown the decision tree fully
  2. Try trimming (pruning) the sub-tree of decision from bottom to up
  3. If after trimming a sub-tree then the generalization error becomes smaller, replace that sub-tree by leaf-node

UNIVERSITY OF ARKANSAS

Pruning tress

“Real life” example of an optimally pruned Decision Tree:



Pruning algorithms are developed and applied on individual trees

optimally pruned single trees are not necessarily optimal in a forest !

actually they tend to be TOO big when boosted, no matter how hard you prune!

Discussions

Advantages of decisions trees:

- Are simple to understand and easy to interpret
- Have value even with small data size to gain important insights
- Help determine the worst, the best and expected values for different scenarios
- Can be easily generated to more advanced methods, such as random forest and gradient boosting

Disadvantages of decision trees:

- Unstable --- noise sensitive
- Relatively inaccurate due to biases or high dimensions
- Calculations can be complex due to high dimensions, data uncertain and correlated outcomes.
- Does not work well for non-rectangular regions (linear restriction)