STUTERN

Decision Tree

SGA07_DATASCI

3rd March 2020

Module Overview

- Concept Learning Recap
- Decision Tree Induction
- Attribute Selection
  - Information Gain
- Tree Pruning

Book Keeping

- Group task submission: Submission moved to 6:00pm on 6th March
- Catch up Live sessions
  - Tuesday: 4 - 6pm
  - Wednesday: 4 - 6pm
  - Thursday: 2 - 4pm

Group Task Submission

- Each team should create a Dropbox or Google drive or Github repo with the following:
- Collection of datasets (raw and clean)
- Scripts (either R or Python)
- A final report that is structured
    - Title
    - Contributors (Team members)
    - Background / Motivation
    - Overview of the data set (You can include any preprocessing methods here)
    - Models (If any was applied)
    - Visualisation (either exploration or prediction)
    - References

Outcome

After this Module, you will;

- Get a recap on concept learning techniques (Find-S & List-then-Eliminate)
- Learn the basic concepts of decision tree as a classification technique
- Cover technical approaches to use information gain for attribute selection
- Understand how to evaluate and prune a decision tree model

Concept Learning Recap

- An initial approach to classification based on inductive learning system that reveals the trade-off between expressiveness and bias
- Find-S and List-then-Eliminate algorithms to provide conceptual framework to search to hypotheses space
- Always be mindful that this classification method is susceptible to overfitting as it works best on training data

Decision Tree (Background)

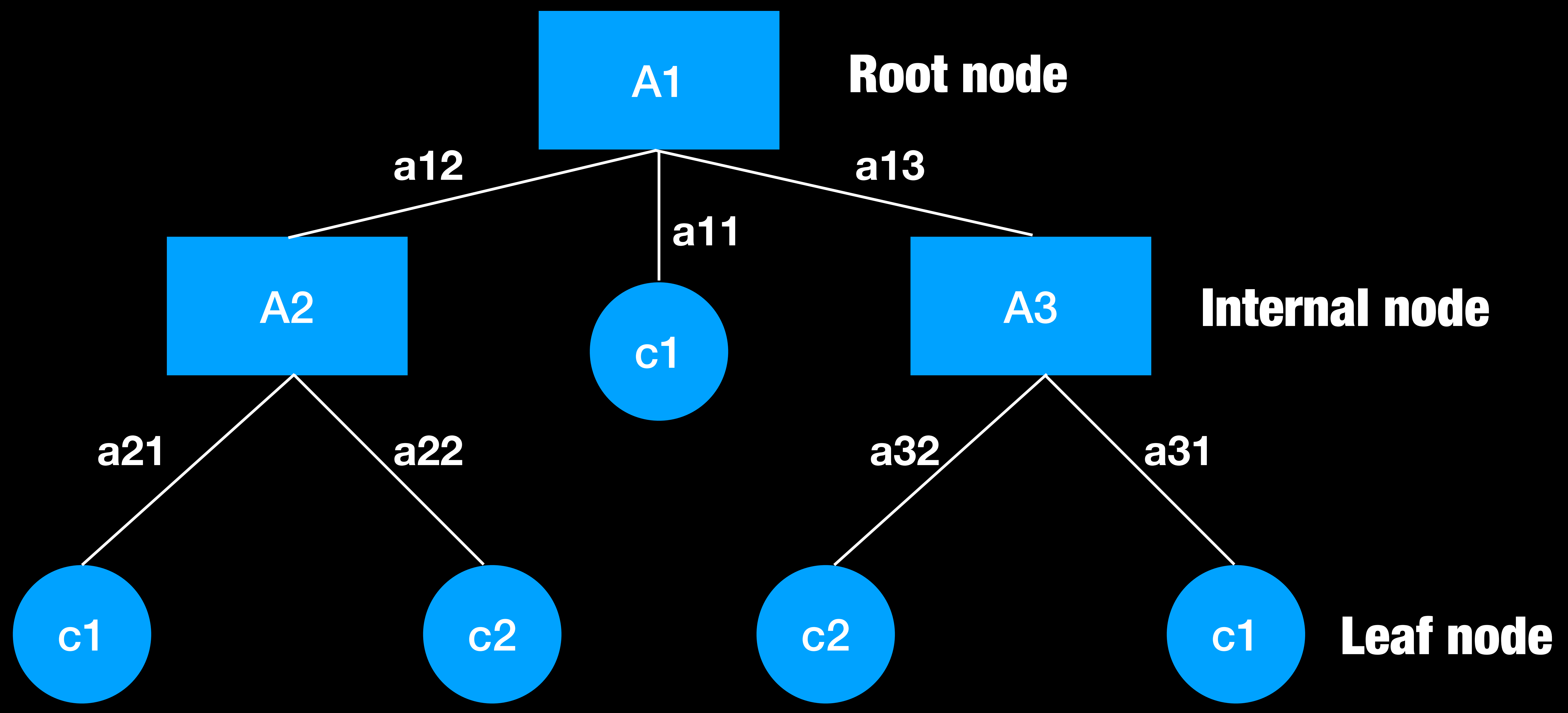
- Data mining technique popularised in the 1980s
- Highly based on human expert systems
- Representation of IF-THEN rules in a flowchart-like structure
- Can be used for qualitative and quantitative class variables

Decision Tree (Def.)

> A decision tree is a flowchart-like tree structure, where each internal node (non-leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label.

STUTERN

Decision Tree (Def.)



Decision Tree (When to use)

- Class attribute is categorical
  - Discretisation when class values are real-valued
- Disjunctive hypothesis space
- Training data may contain errors and/or missing values
- More useful for many real-world classification than concept learning

Top-Down Decision Tree Induction

- Induction consists of two parts
  - Tree construction
    - At start, all training instances are at the root
    - Partition instances recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers
    - Tackle overfitting

Decision Tree (Algorithm)

- Input:
  - Data partition, D, which is a set of training tuples and their associated class labels;
  - attribute list, the set of candidate attributes;
  - Attribute selection method, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a
    splitting attribute and, possibly, either a split-point or splitting subset.

Decision Tree (Algorithm)

- Method:
  - create a node N ;
  - if tuples in D are all of the same class, C, then
    - return N as a leaf node labeled with the class C;
  - if attribute list is empty then
    - return N as a leaf node labeled with the majority class in D; // majority voting
  - apply Attribute selection method(D, attribute list) to find the “best” splitting criterion;
  - label node N with splitting criterion;

Decision Tree (Algorithm)

- Method (contd.):
  - if splitting attribute is discrete-valued and
    - multiway splits allowed then // not restricted to binary trees
    - attribute list ← attribute list − splitting attribute; // remove splitting attribute
  - for each outcome j of splitting criterion
    // partition the tuples and grow subtrees for each partition
    - let Dj be the set of data tuples in D satisfying outcome j; // a partition
    - if Dj is empty then
      - attach a leaf labeled with the majority class in D to node N ;
    - else attach the node returned by Generate decision tree(Dj, attribute list) to node N;
  - endfor
  - returnN;

STUTERN

Attribute Selection Options

- Arbitrary
- Information Gain
- Gini Index

> An attribute selection measure is a heuristic for selecting the splitting criterion that "best" separates a given data partition, D, of class-labeled training tuples into individual classes.

Information Gain (Background)

- First used in ID3 algorithm devised by Ross Quinlan in 1978.
- Widely used in many different data mining applications:
  - Medical
  - Fraud detection
  - ‘Churn’ reduction
- At each tree induction iteration, splitting on any attribute has property that average entropy of resulting subsets will be less than (or equal to) that of previous set.

Information Gain (Procedure)

- At each node, entropy calculated for each remaining attribute
- Attribute with highest information gain (i.e. greatest entropy reduction) chosen as splitting attribute

Information Gain (Maths)

- Assume that using attribute A with v values, set D will be partitioned into sets {D1, D2 , ..., Dv}:
  - The expected information needed to classify a tuple in D is given by

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

STUTERN

Information Gain (Maths)

- Assume that using attribute A with v values, set D will be partitioned into sets {D1, D2 , ..., Dv}:
  - If D_i contains p_i instances of P and n_i instances of N, entropy, or expected information needed to classify instances in all subtrees D_i is

$$Info_A(D) = -\sum_{j=1}^{v} \frac{p_i + n_i}{p + n} \times Info(D_j)$$

Information Gain (Maths)

- Assume that using attribute A with v values, set D will be partitioned into sets {D1, D2 , ..., Dv}:
  - Information gain by splitting on A:

$$Gain(A) = Info(D) - Info_A(D)$$

Access Bank

A bank loans officer needs analysis of her data to learn which loan applicants are "safe"

and which are "risky" for the bank.

Access Bank Data

ID	Name	Age	Income	Loan
1	Bukola Saraki	Youth	Low	Risky
2	Segun Obasanjo	Youth	Low	Risky
3	Goodluck Jonathan	Middle_aged	High	Safe
4	Muhammad Buhari	Middle_aged	Low	Risky
5	Godwin Emefiele	Senior	Low	Safe
6	Babatunde Fashola	Senior	Medium	Safe
7	Mojisola Adeyeye	Middle_aged	High	Safe

Access Bank Dataset

- Two target classes; risky and safe
- Out of 7 instances, 3 classified risky, 4 safe

$$Info(Risky) = -(\frac{3}{7})log_2\frac{3}{7} = 0.5239 \qquad Info(Safe) = -(\frac{4}{7})log_2\frac{4}{7} = 0.4613$$

$$Info(D) = 0.5239 + 0.4613 = 0.9852$$

Access Bank Dataset

- Three levels for Age attribute; youth, middle_aged and senior
- For Youth, 2 classified as risky, 0 safe
- For Middle_aged, 1 classified as risky, 2 safe
- For Senior, 0 classified as risky, 2 safe

$$Info_{age}(D) = \frac{2}{7} \times (-\frac{2}{2}log_2\frac{2}{2}) + \frac{3}{7} \times (-\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3}) + \frac{2}{7} \times (-\frac{2}{2}log_2\frac{2}{2}) = 0.3936$$

$$Gain(Age) = 0.9852 - 0.3936 = 0.5916$$

Access Bank Dataset

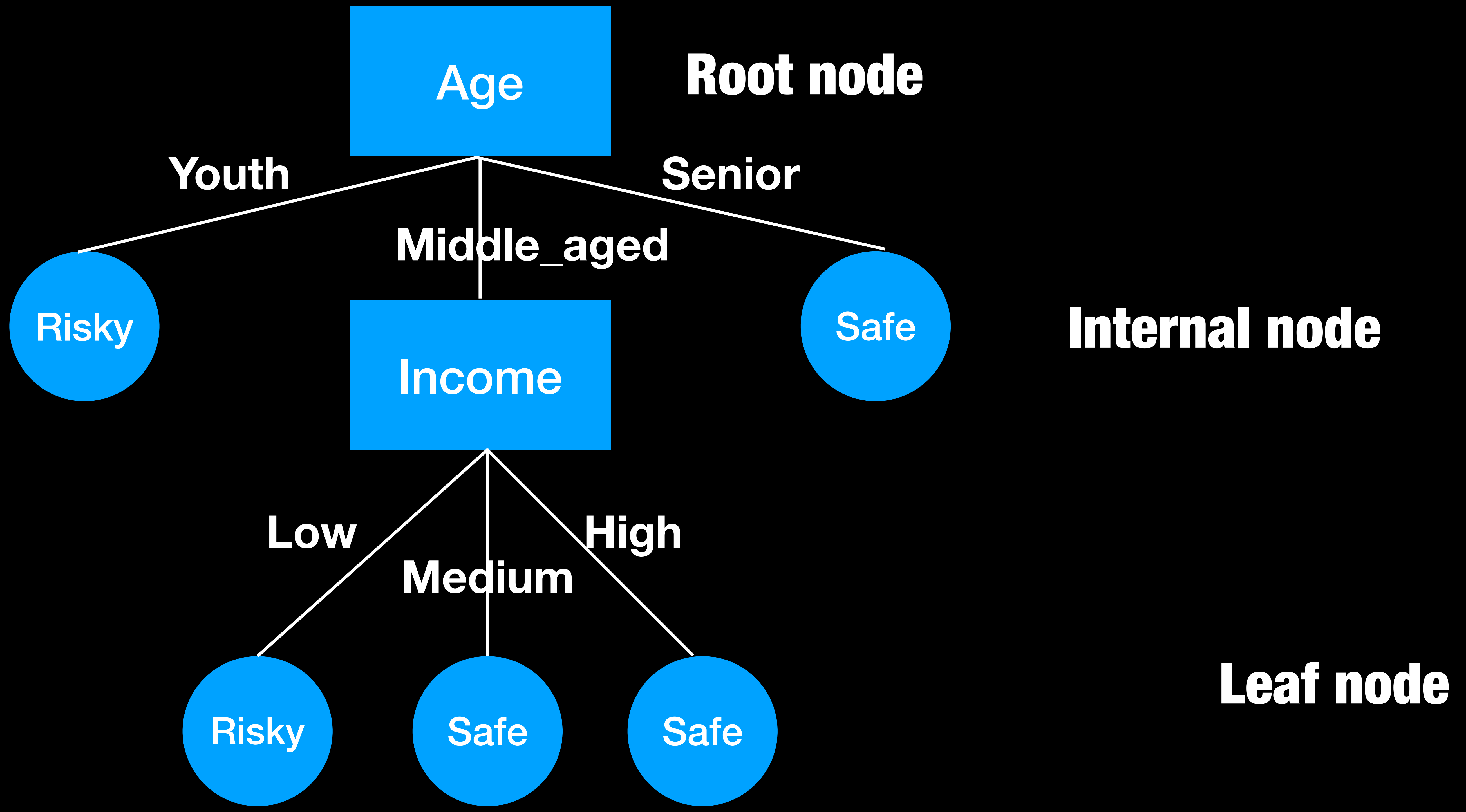
- Three levels for Income attribute; low, medium and high
- For Low, 3 classified as risky, 1 safe
- For Medium, 0 classified as risky, 1 safe
- For High, 0 classified as risky, 2 safe

$$Info_{income}(D) = \frac{4}{7} \times (-\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4}) + \frac{1}{7} \times (-\frac{1}{1}log_2\frac{1}{1}) + \frac{2}{7} \times (-\frac{2}{2}log_2\frac{2}{2}) = 0.4636$$

$$Gain(Age) = 0.9852 - 0.4636 = 0.5216$$

STUTERN

Access Bank Dataset



Comparing Attribute Selection Options

- Information Gain
  - Biased towards splitting on multi-valued attributes
- Gini Index
  - Biased towards splitting on multi-valued attributes
- Gain Ratio
  - Reduces bias in Information Gain/Gini Index
- Best choice will often depend on data

Dealing with Decision Tree Clashes

- Delete branch:
  - Similar to removing instances in clash set from training set
- Majority voting:
  - Similar to changing instance labels in training set
- Clash threshold:
  - Assign class of most common class of clash instances if proportion ≥ clash threshold
  - Discard clash instances and corresponding branch if not

Tree Quality Measures

- Speed and scalability:
  - Time to construct model
  - Time to use model
- Interpretability:
  - Understanding and insight provided by model
- 'Goodness' of rules:
  - Decision tree size
  - minimum description length
- Accuracy:
  - How many unseen instances correctly classified?

Decision Tree Pruning

- After pruning, tree will be smaller and simpler:
  - At least as accurate
  - Fewer branches
- Two basic approaches:
  - Pre-pruning applied as tree learned
  - Post-pruning applied after tree learned

Decision Tree Pruning

- Pre-pruning:
  - Do not split if result is quality measure falling below threshold
- Post-pruning:
  - Remove branches from full tree to create set of progressively pruned trees
  - Vary thresholds and use validation dataset to decide on best pruned tree

Pre-pruning

- Apply terminating condition to decide when to stop tree development.
- Size cut-off:
  - Prune if sub-tree contains fewer than threshold number of instances
- Maximum depth cut-off:
  - Prune if branch length exceeds branch length threshold/MDL

Post-pruning

- Convert tree to set of IF-THEN rules
  - Generalise each rule by removing some conditions
  - Sort pruned rules by estimated accuracy
- Reduced error:
  - Calculate error if pruned and prune if less than current error

Practice Lab

Implement a decision tree classification model in R

Use the following Instructions:

- Use the Iris Dataset in R
- Explore the dataset to get some intuition
- Partition your data into train and test sets
- Build your decision tree model using 'party' package
- Evaluate your model on the test set
- Explore pruning your model on minimum node split

Recap/Summary

At the end of this Module, you should understand;

- Get a recap on concept learning techniques (Find-S & List-then-Eliminate)
- Learn the basic concepts of decision tree as a classification technique
- Cover technical approaches to use information gain for attribute selection
- Understand how to evaluate and prune a decision tree model

STUTERN

Suggested Material

- Machine Learning by Tom Mitchell Chapter 3 Pages 52 - 80
- Data Mining Concepts and Techniques (3rd Edition) by Jiawei Han, Micheline Kamper and Jian Pei: Chapter 8 (Section 2) Pages 330 - 348
- https://en.proft.me/2016/11/9/classification-using-decision-trees-r/
- https://www.youtube.com/watch?v=RmajweUFKvM