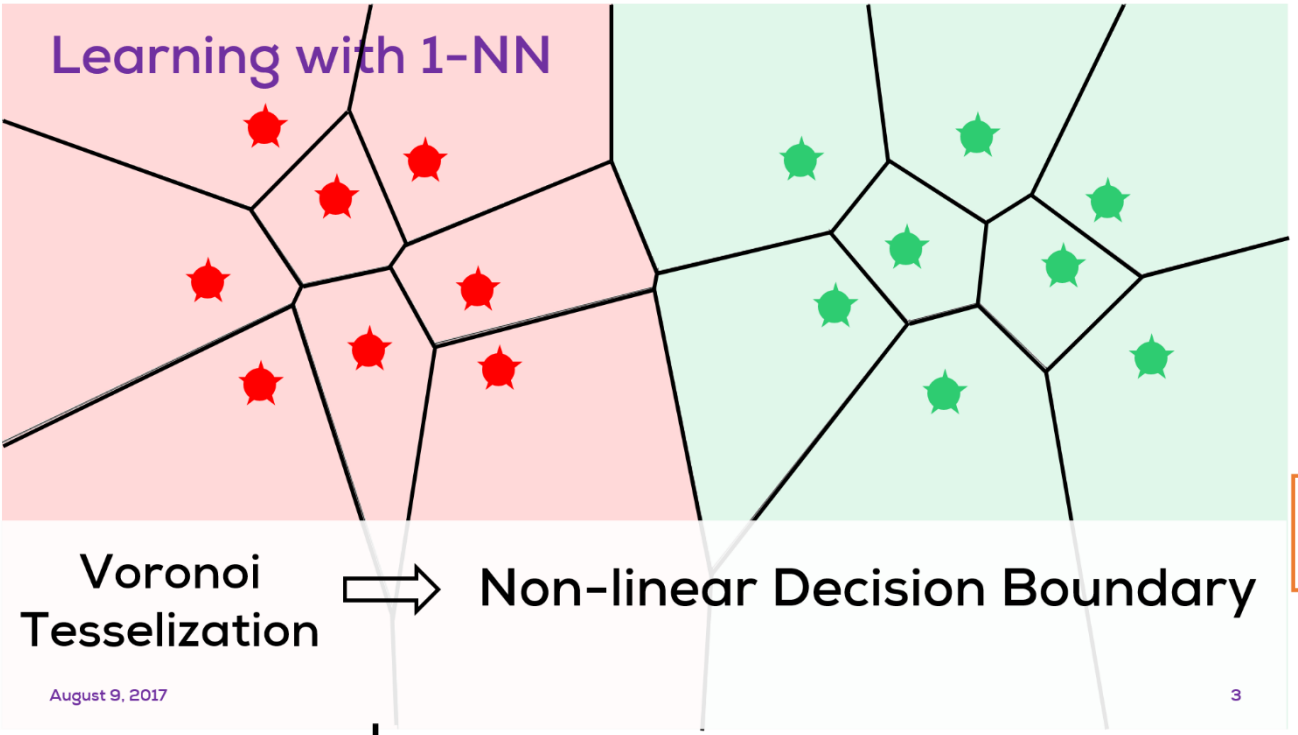
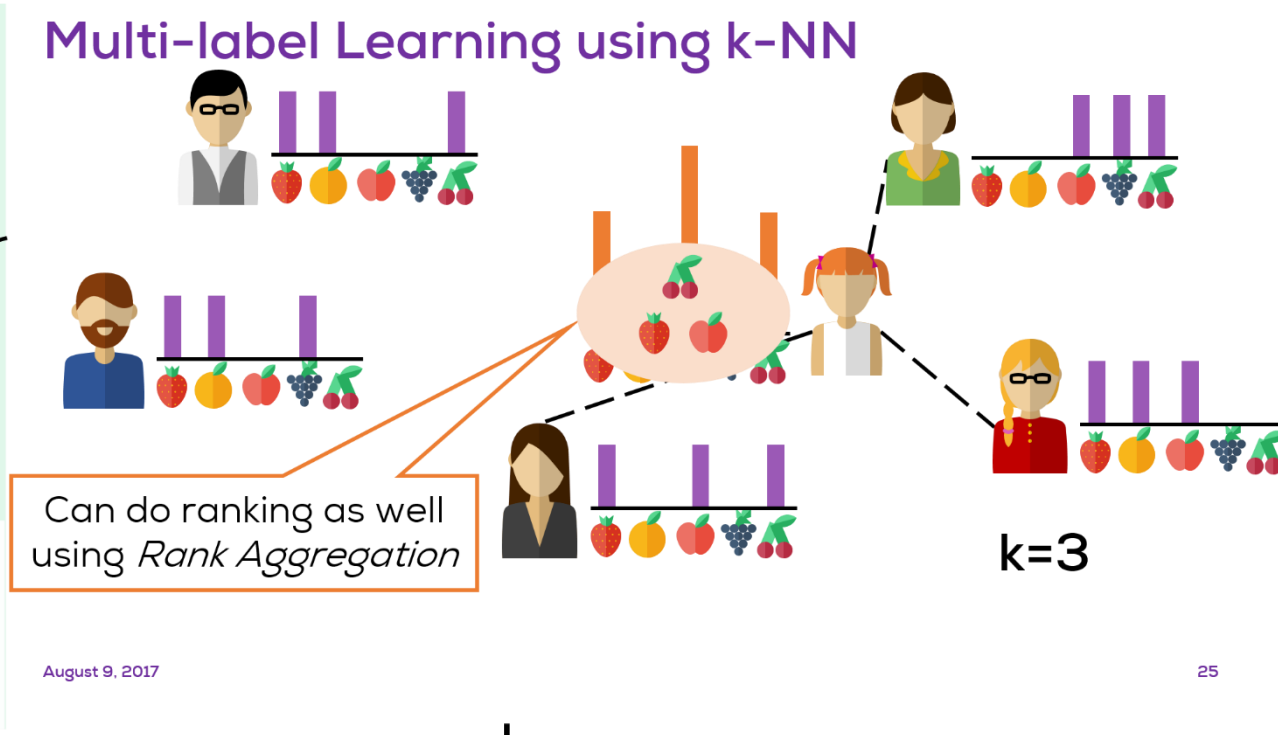
Local Methods-II

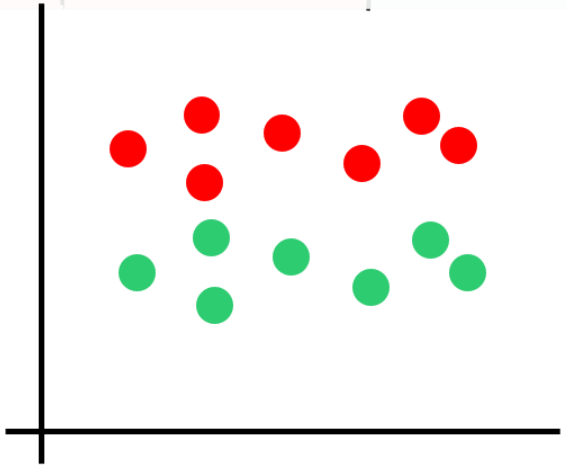
CS771: Introduction to Machine Learning

Purushottam Kar

Recap

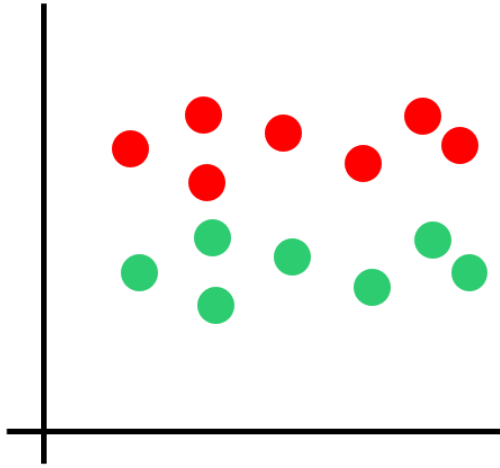




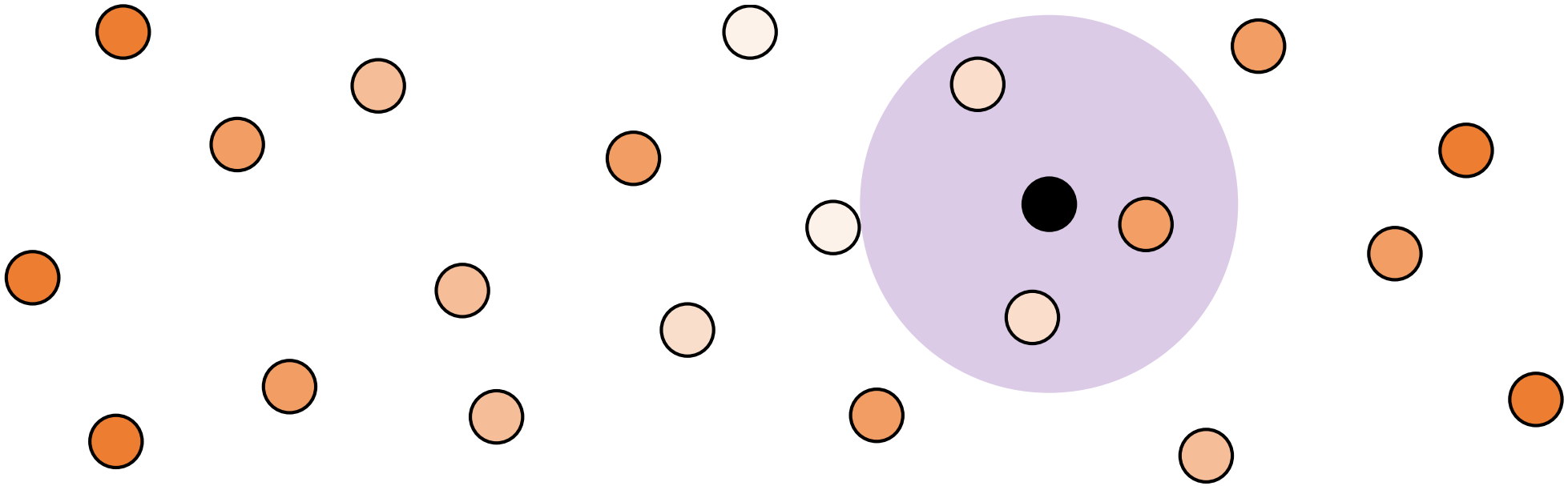


$$d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^\top M (\mathbf{x} - \mathbf{y})$$

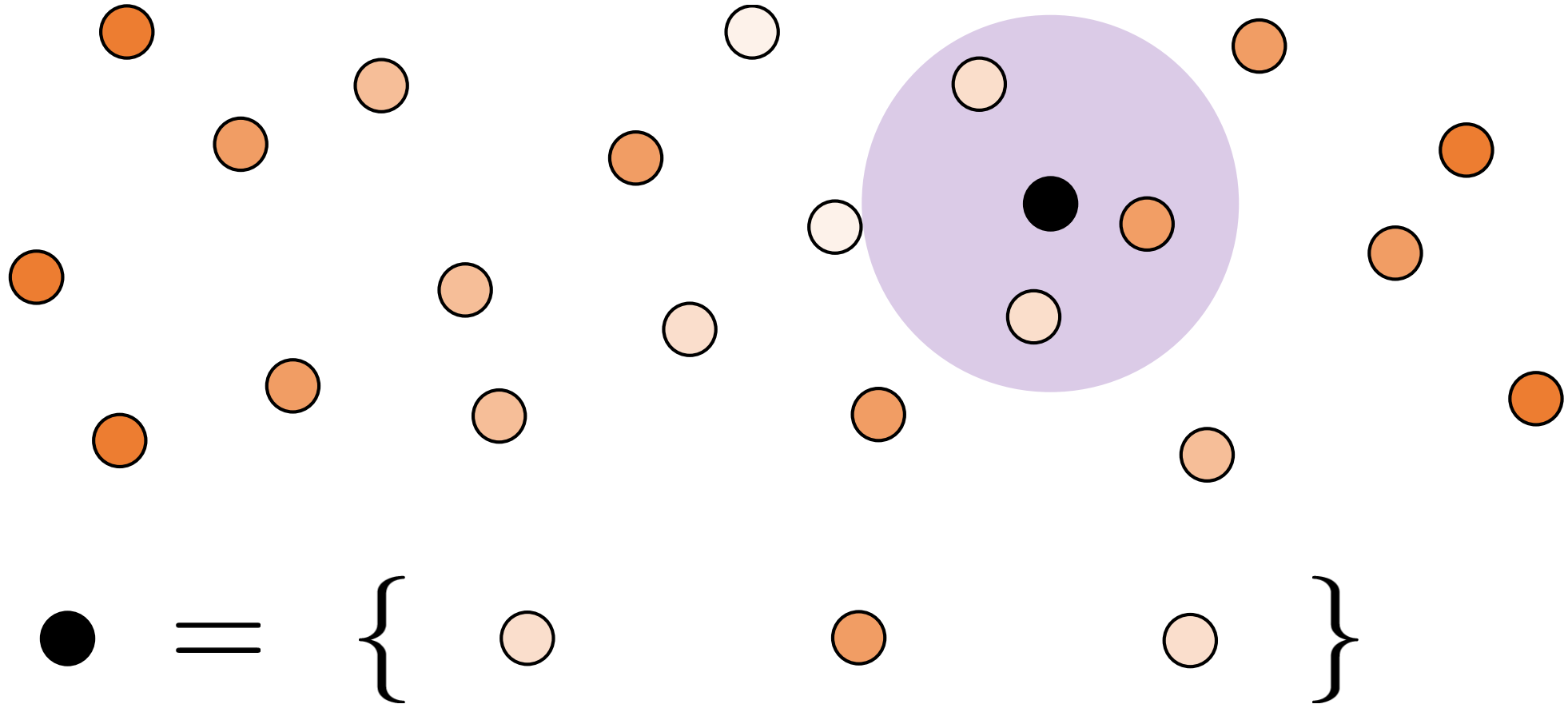
$$M = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$



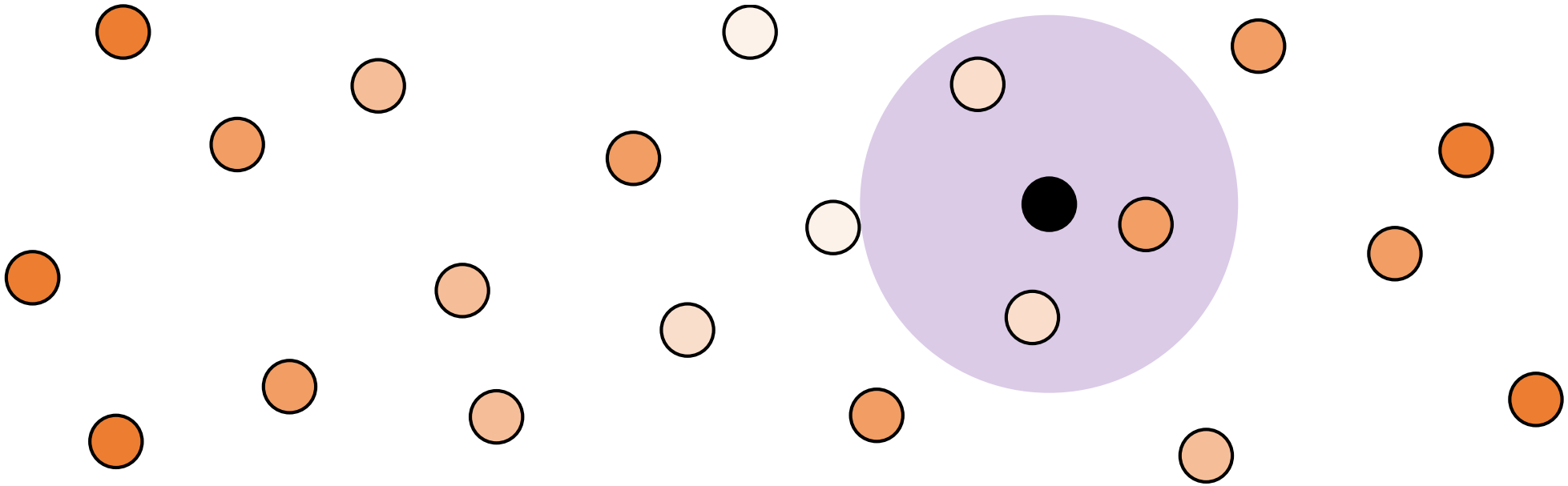
Regression with r-NN



Regression with r-NN

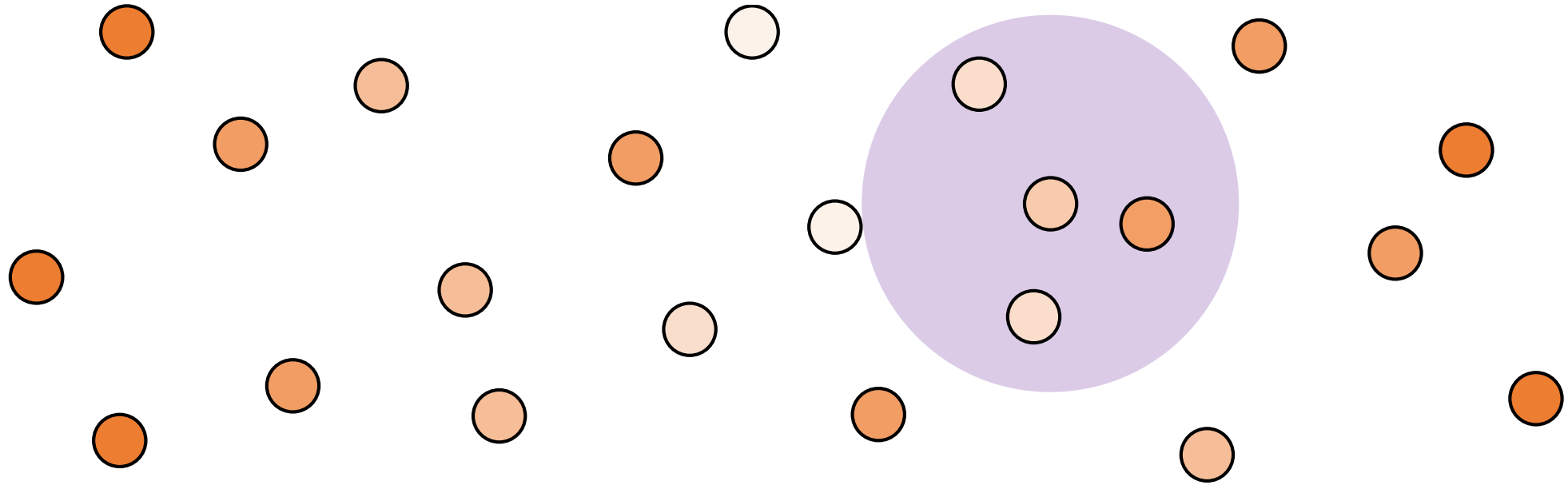


Regression with r-NN



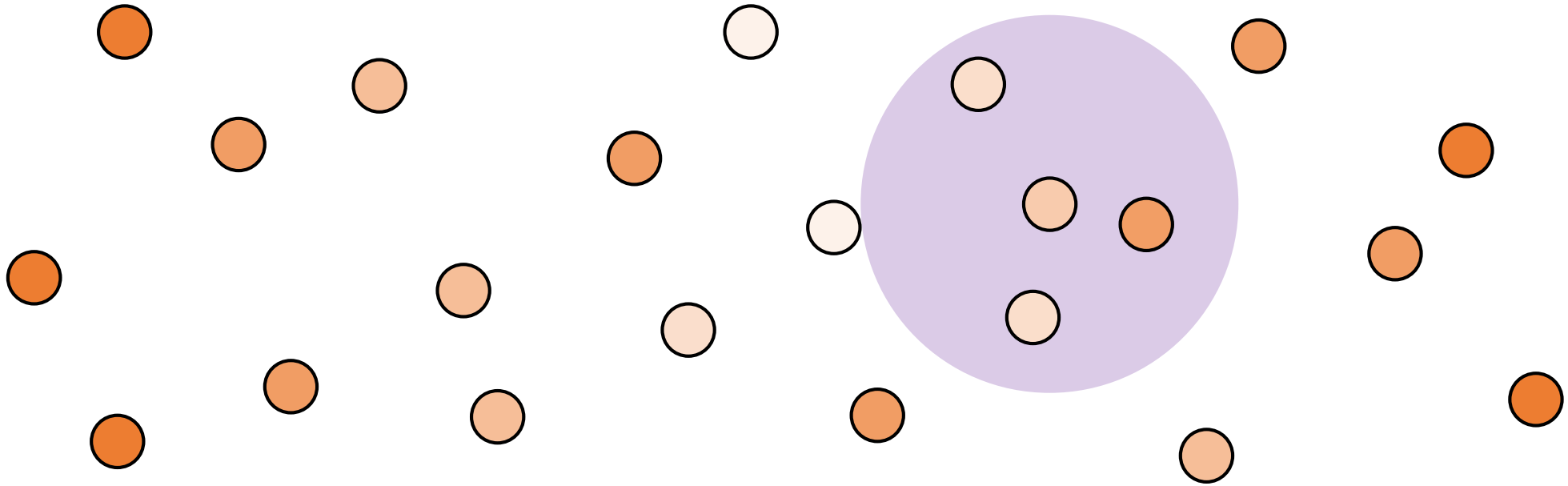
$$\bullet = \left\{ \frac{1}{3} \circ + \frac{1}{3} \circ + \frac{1}{3} \circ \right\}$$

Regression with r-NN



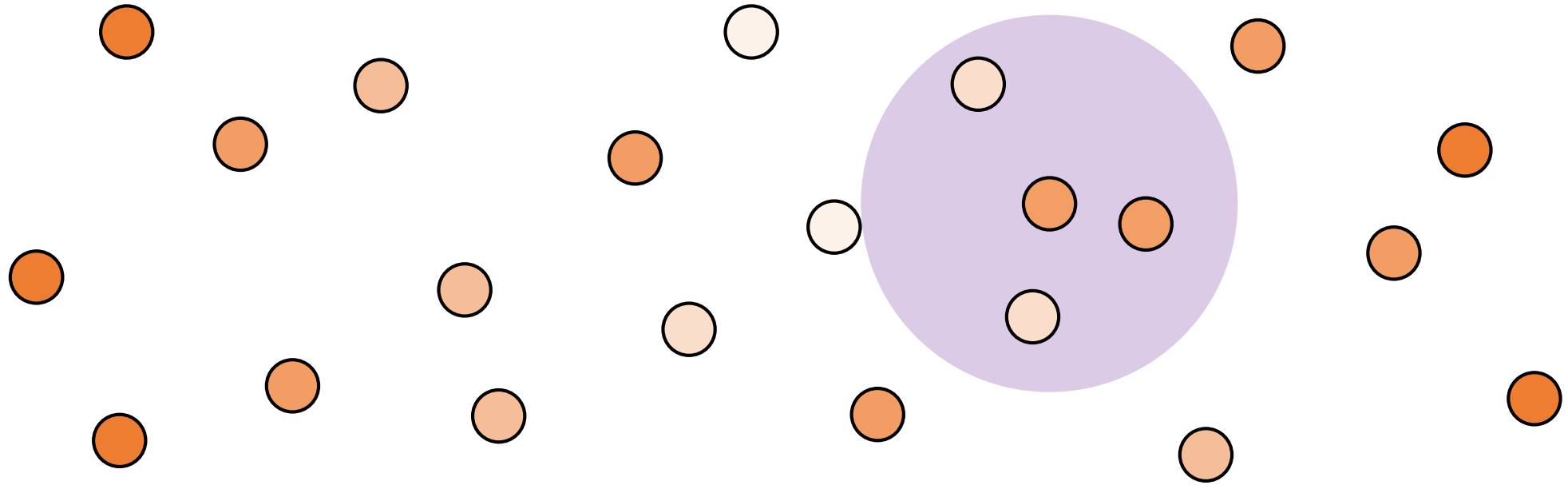
$$\circ = \left\{ \frac{1}{3} \circ + \frac{1}{3} \circ + \frac{1}{3} \circ \right\}$$

Regression with Weighted r-NN



$$\bullet = \left\{ \frac{1}{5} \circ + \frac{3}{5} \bullet + \frac{1}{5} \circ \right\}$$

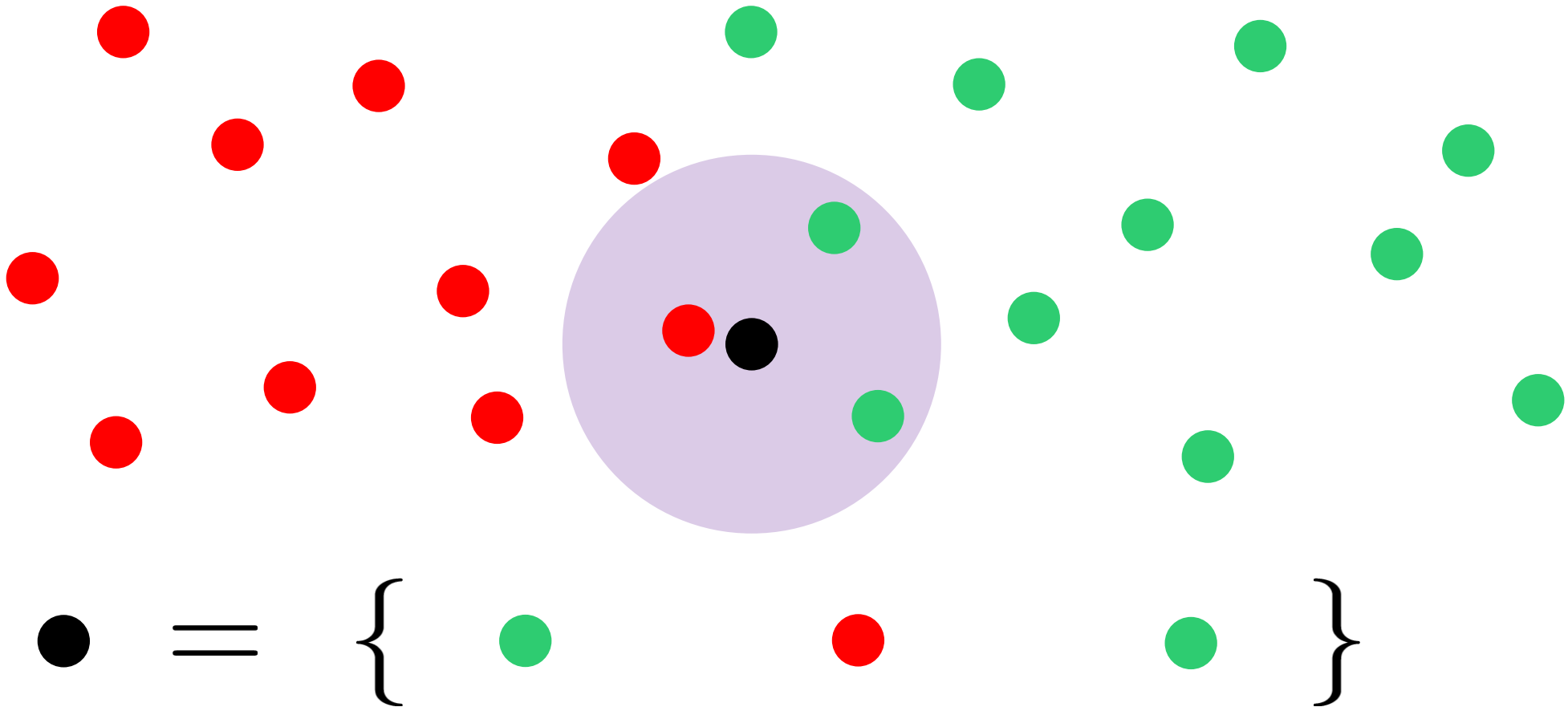
Regression with Weighted r-NN



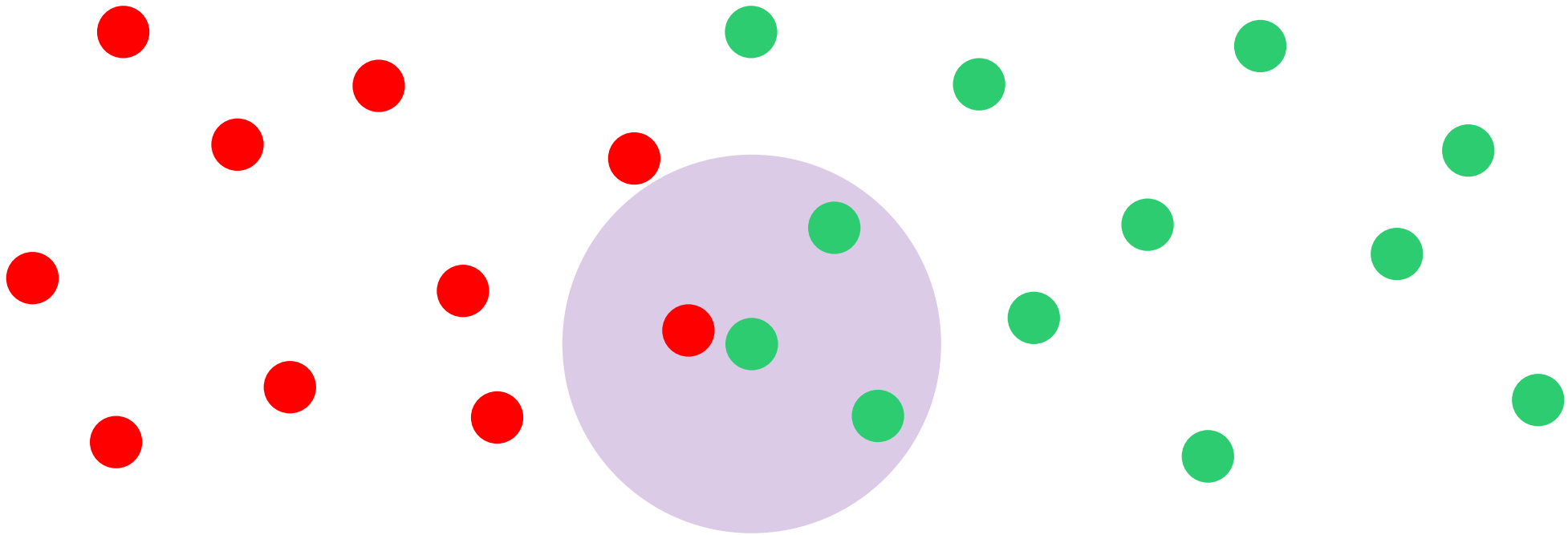
$$\bullet = \left\{ \frac{1}{5} \circ + \frac{3}{5} \bullet + \frac{1}{5} \circ \right\}$$

Can use a similar trick with k-nn too!

Classification with Weighted r-NN

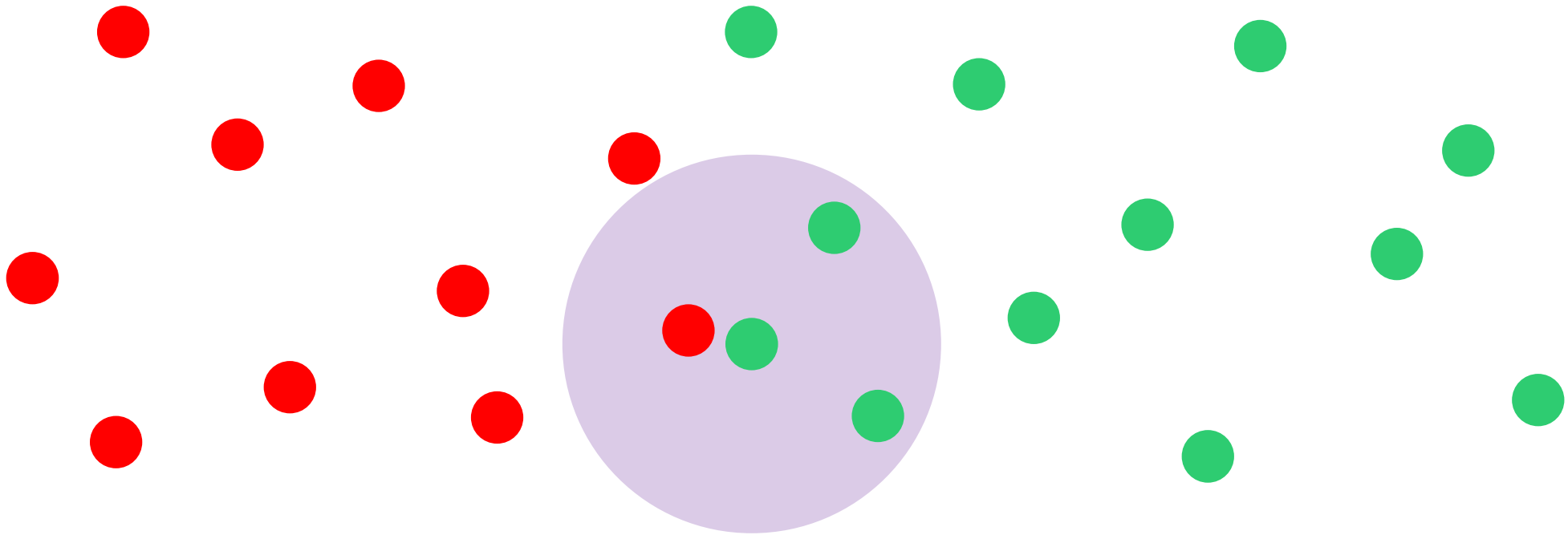


Classification with Weighted r-NN



$$\bullet = \left\{ \frac{1}{3}(+1) + \frac{1}{3}(-1) + \frac{1}{3}(+1) \right\}$$

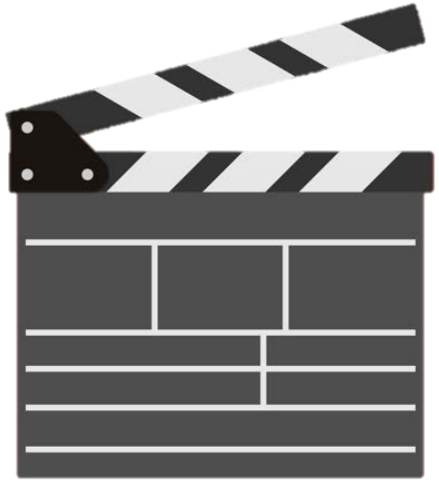
Classification with Weighted r-NN



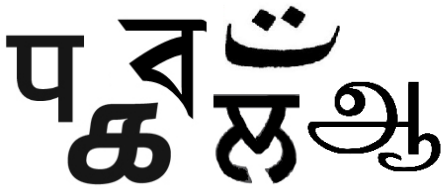
$$\bullet = \left\{ \frac{1}{5}(+1) + \frac{3}{5}(-1) + \frac{1}{5}(+1) \right\}$$

All it takes are 20 questions

Guess the Movie!



?



Original Language

Year of Release

2010,2011,...2017

Box-Office Collection

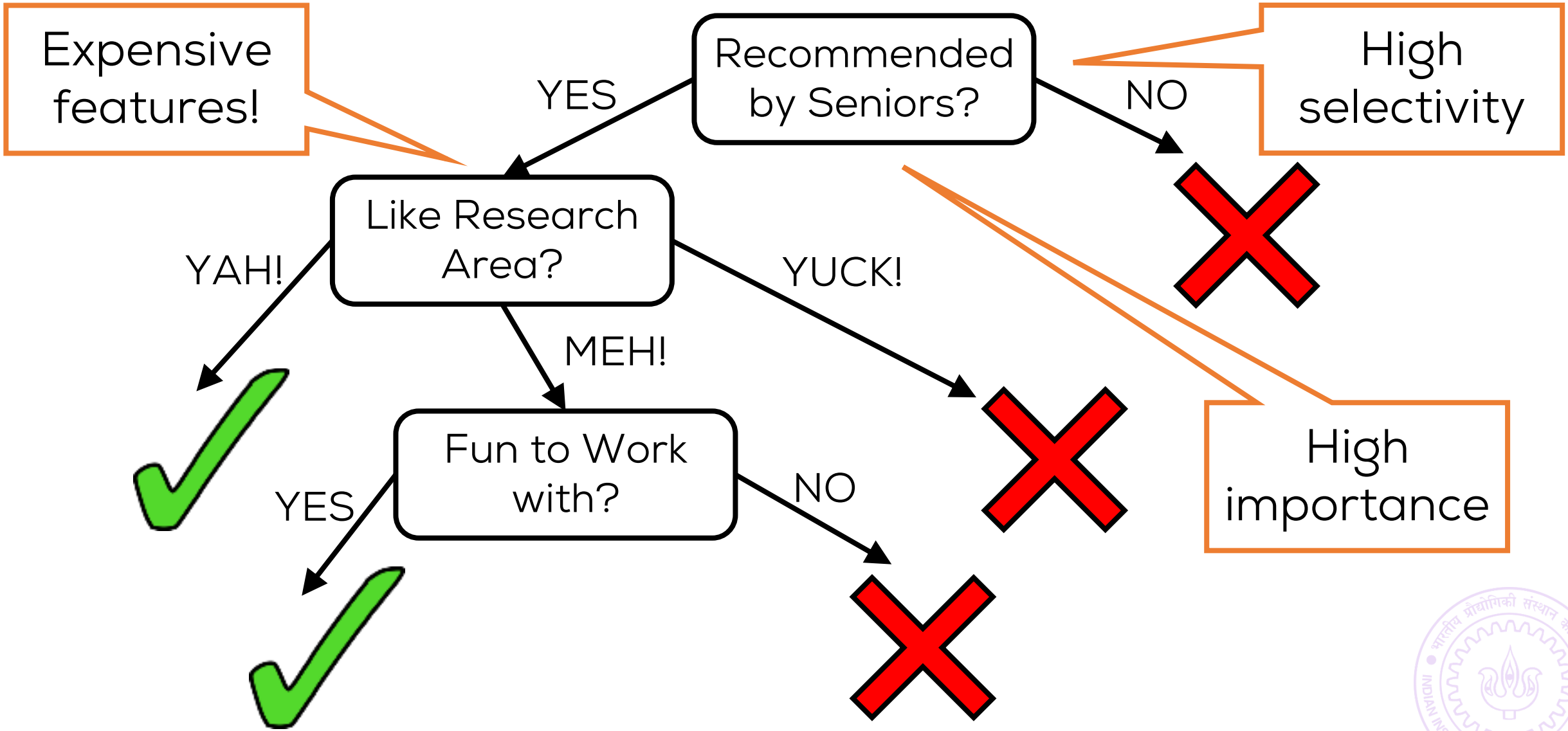
Low (< INR 100 cr)

Medium (INR 100-1000 cr)

High (> 1000 cr)

shutterstock.com, inktalks.com, glamsham.com, youtube.com

Choose an Adviser!

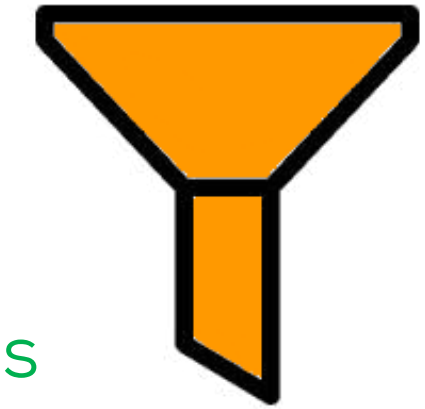


Learning with Decision Trees

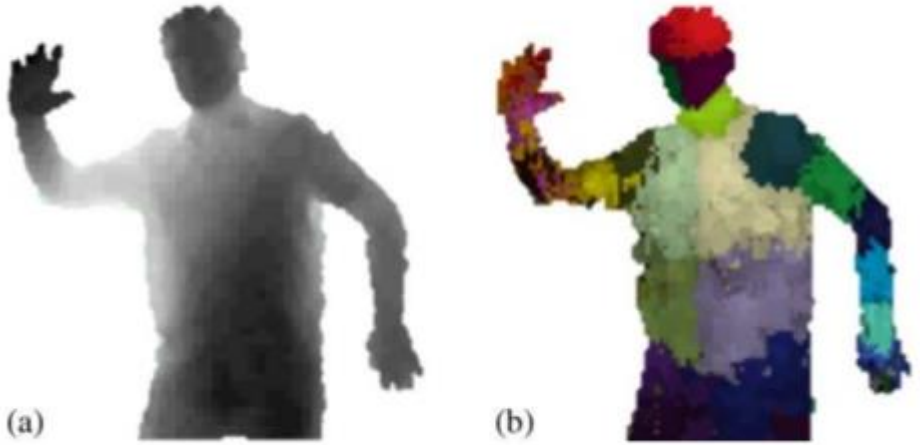
… and Decision Forests

Decision Trees

- Very versatile
- Easy to implement
- Extremely fast at test time
- Easy to interpret by humans
- Can be voluminous
- Can overfit badly
- Very very popular

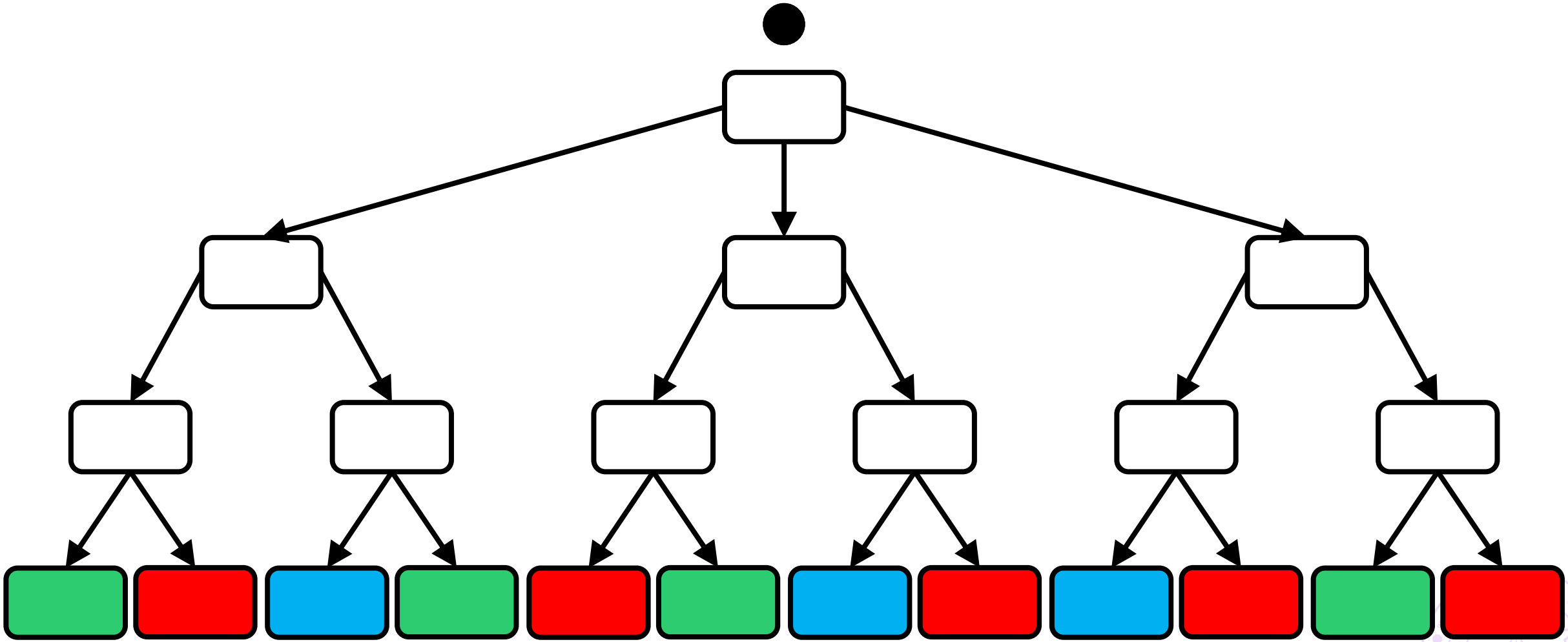


Gaming – Kinect for Xbox 360

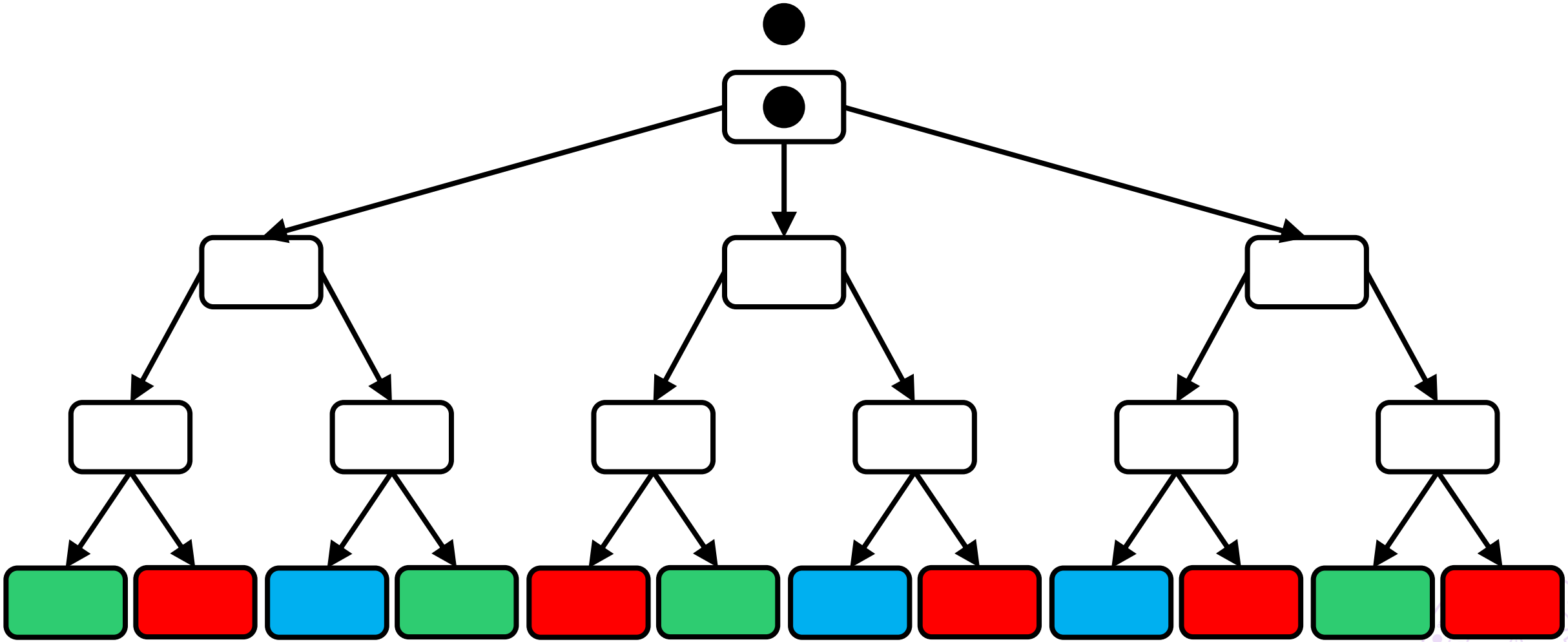


Shotton et al. Real-Time Human Pose Recognition in Parts from a Single Depth Image, CVPR 2011

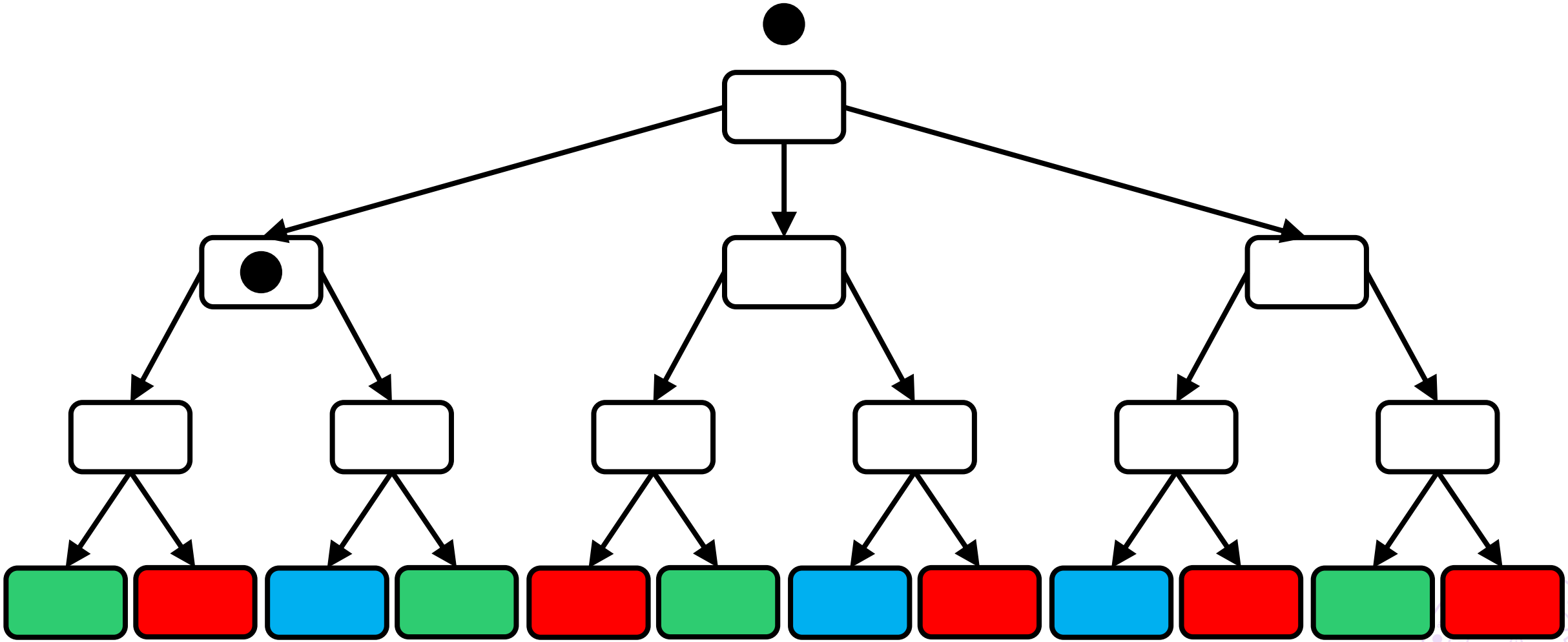
Classification with Decision Trees



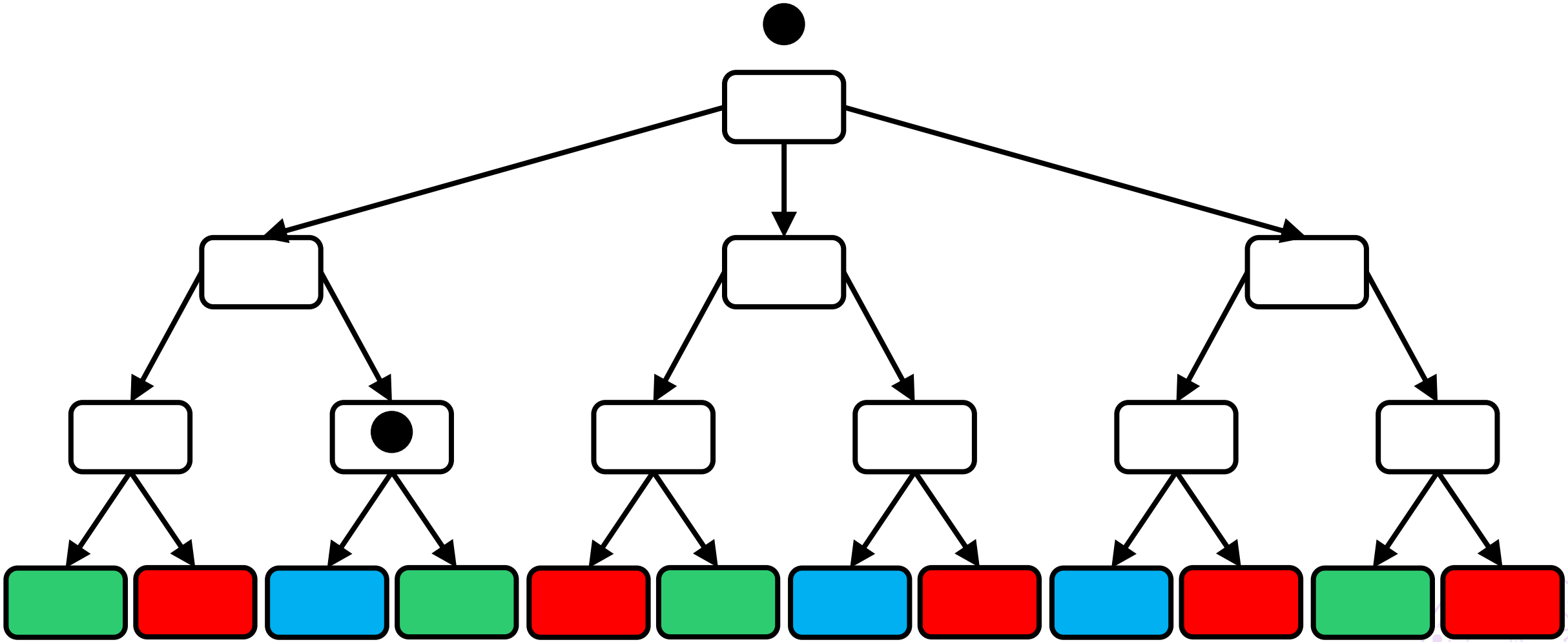
Classification with Decision Trees



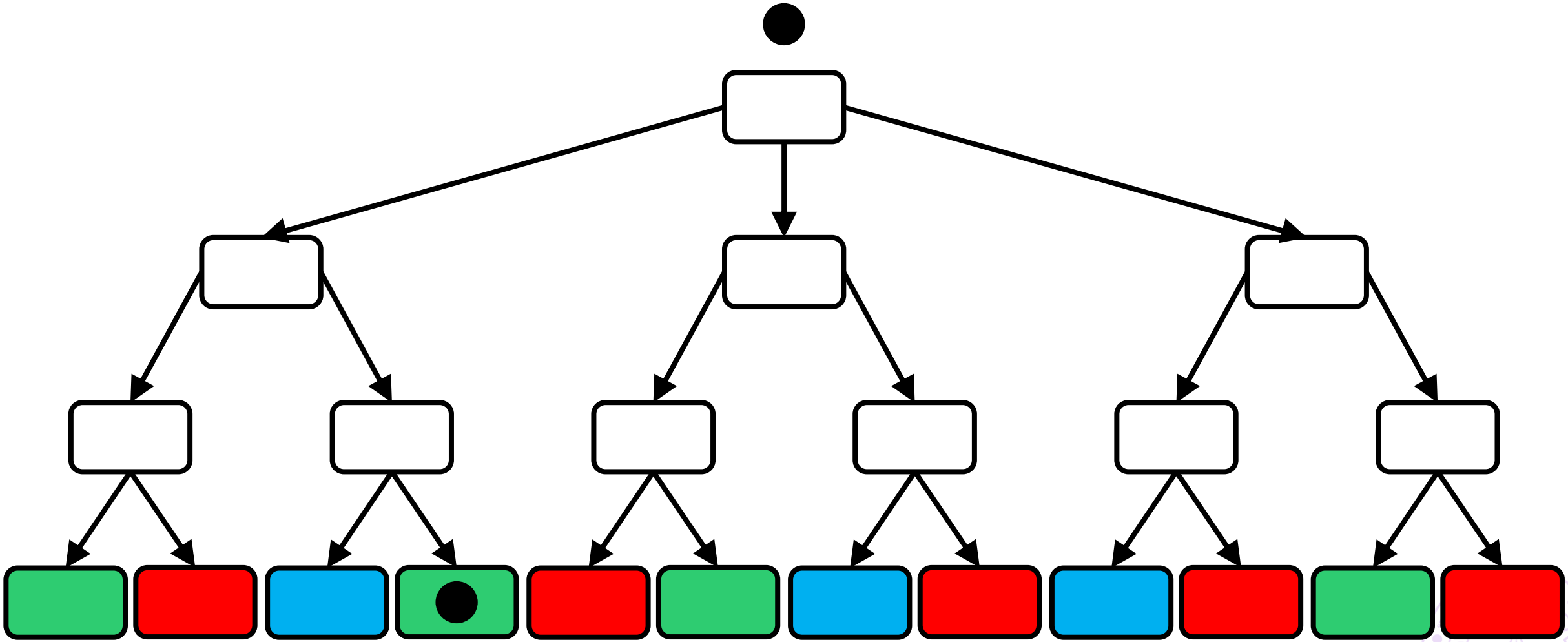
Classification with Decision Trees



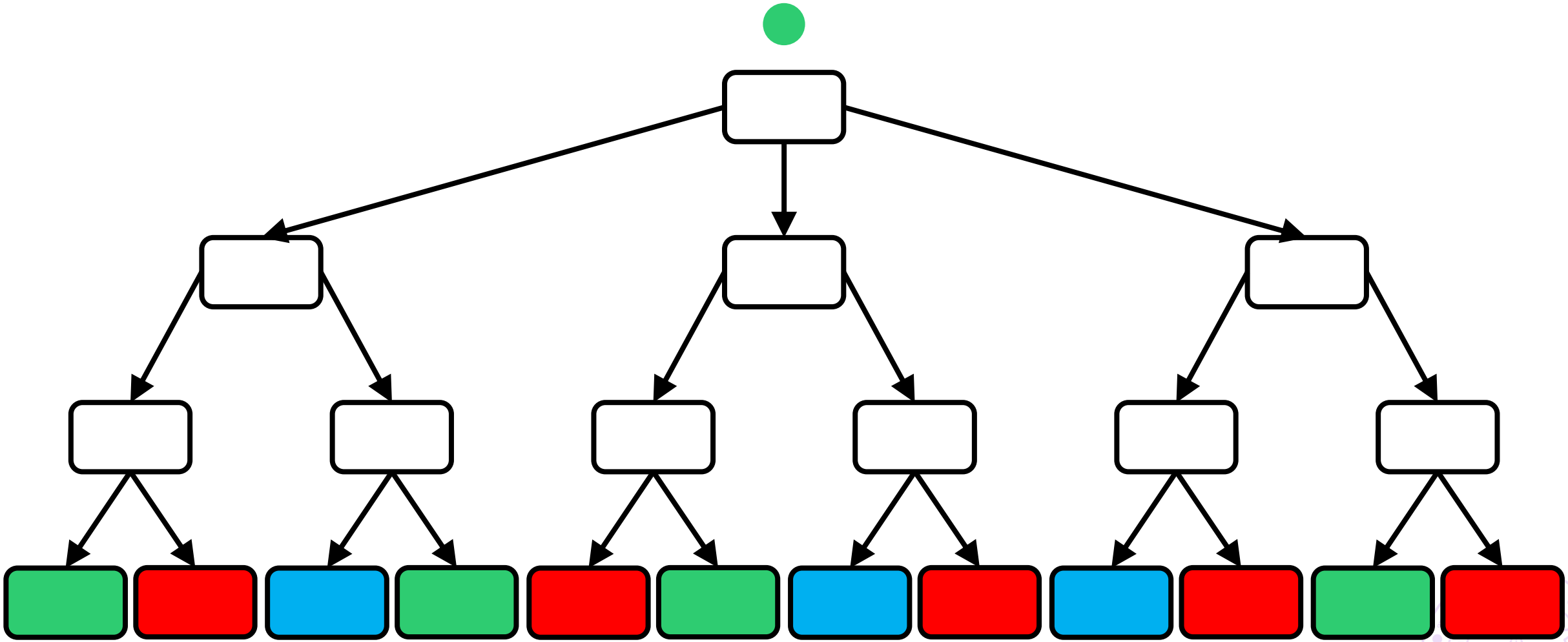
Classification with Decision Trees



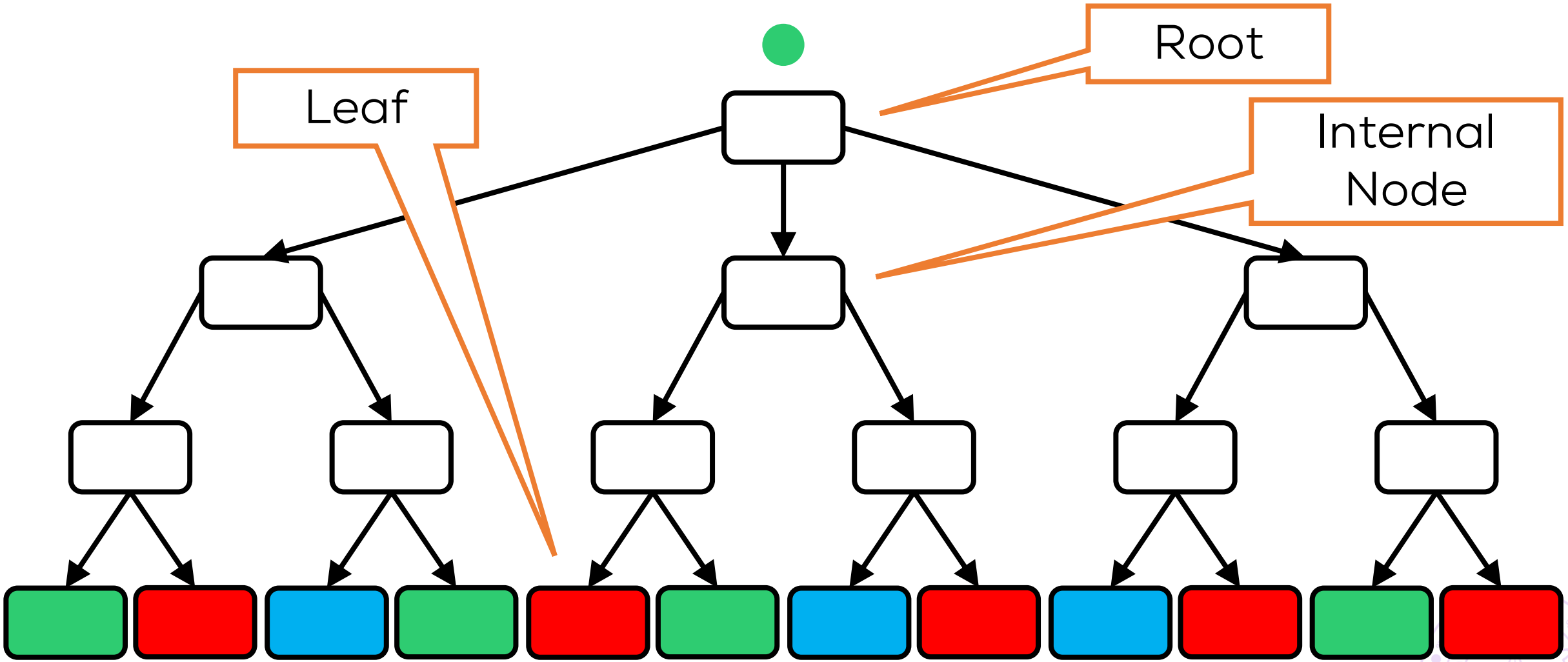
Classification with Decision Trees



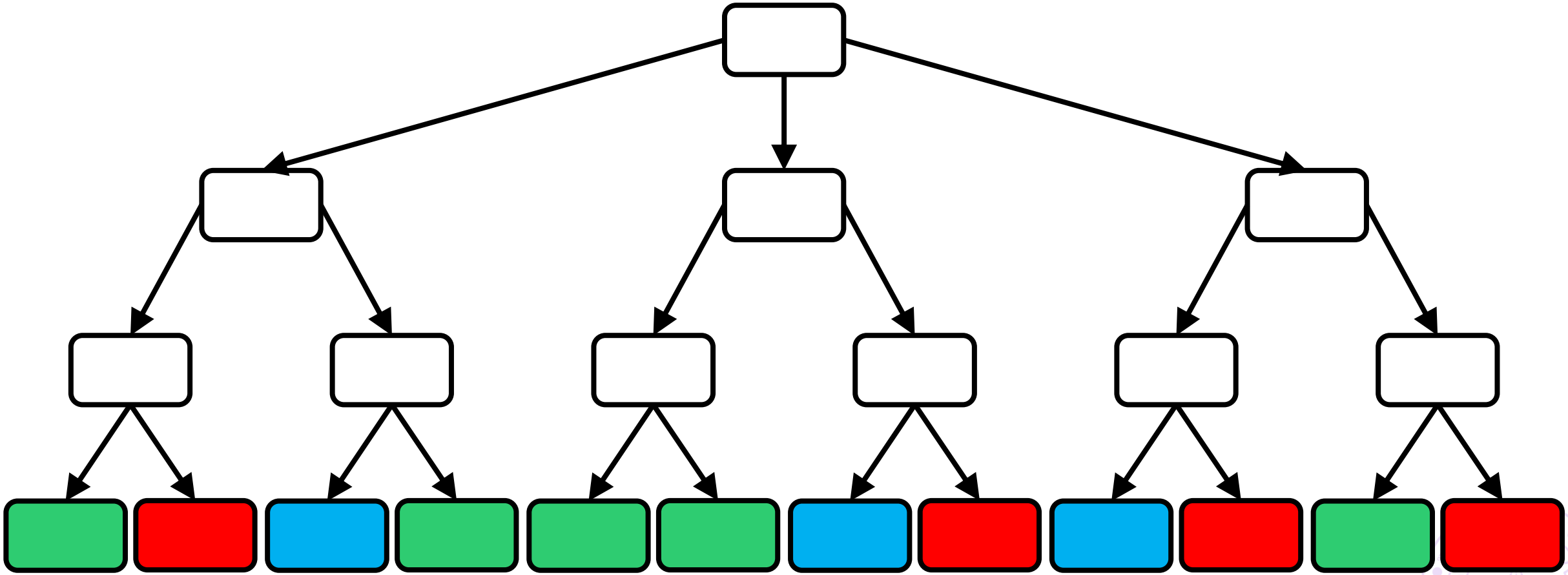
Classification with Decision Trees



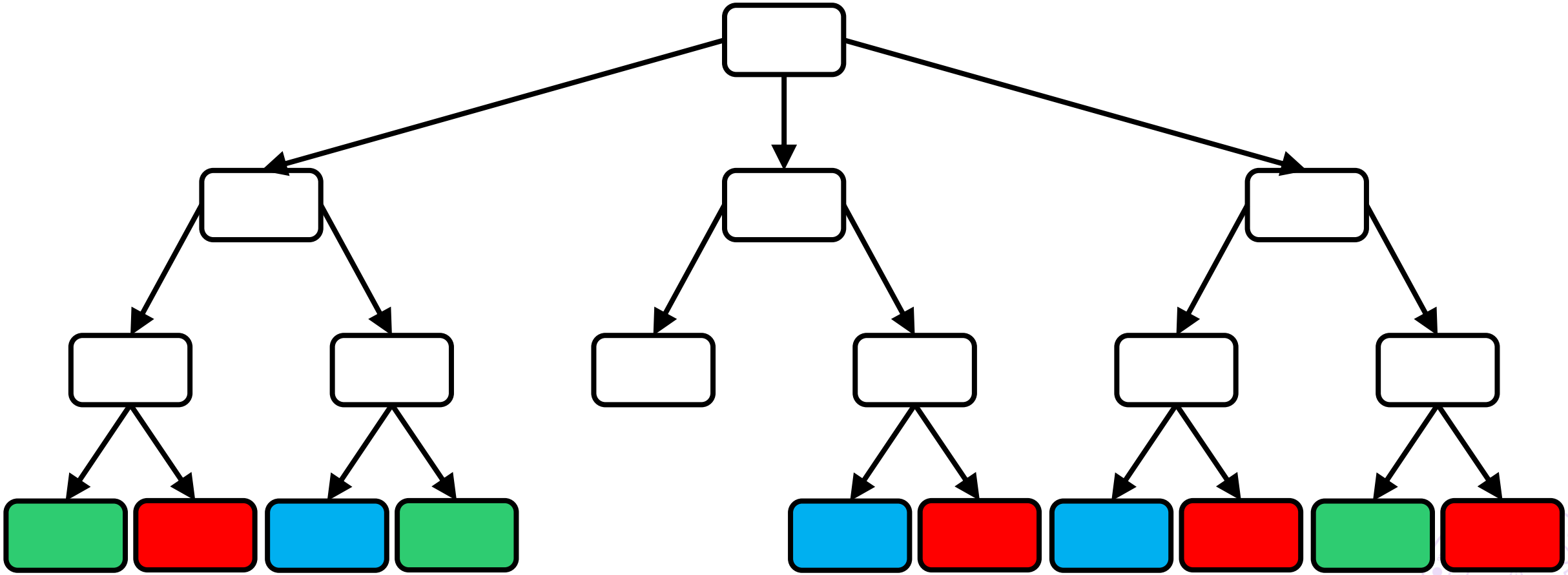
Classification with Decision Trees



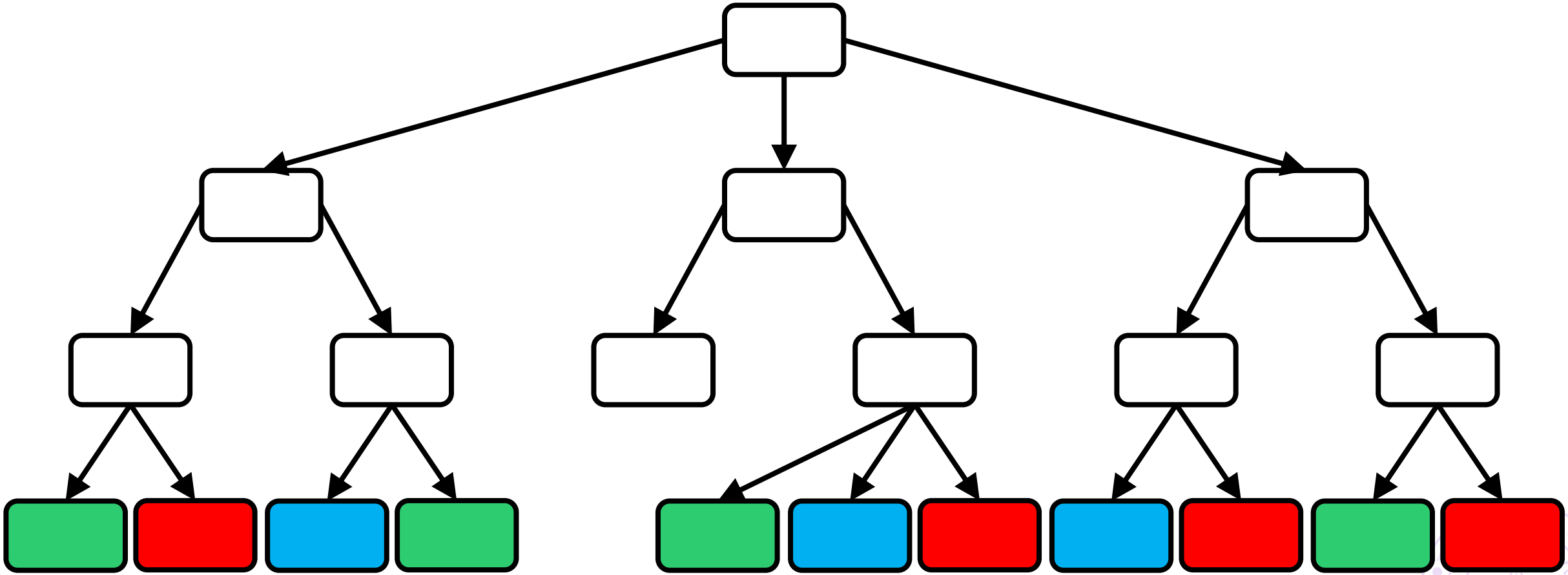
Decision Trees – all shapes and sizes



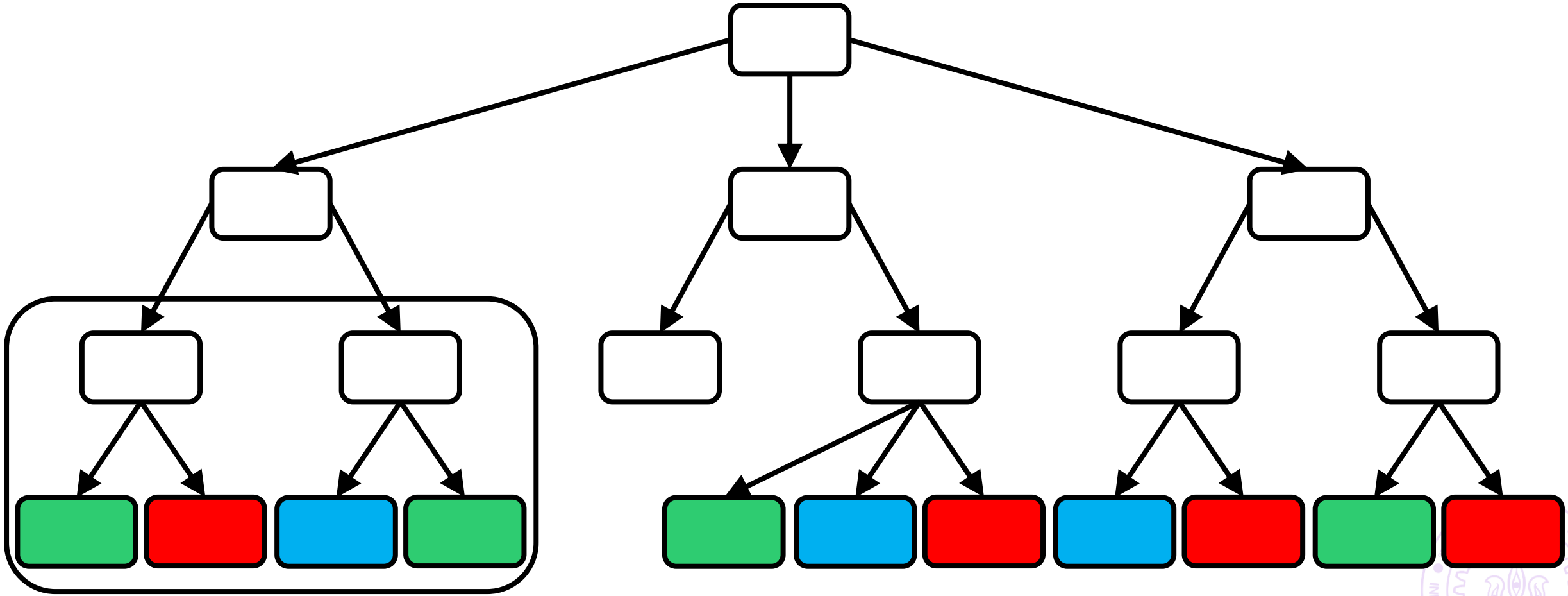
Decision Trees – all shapes and sizes



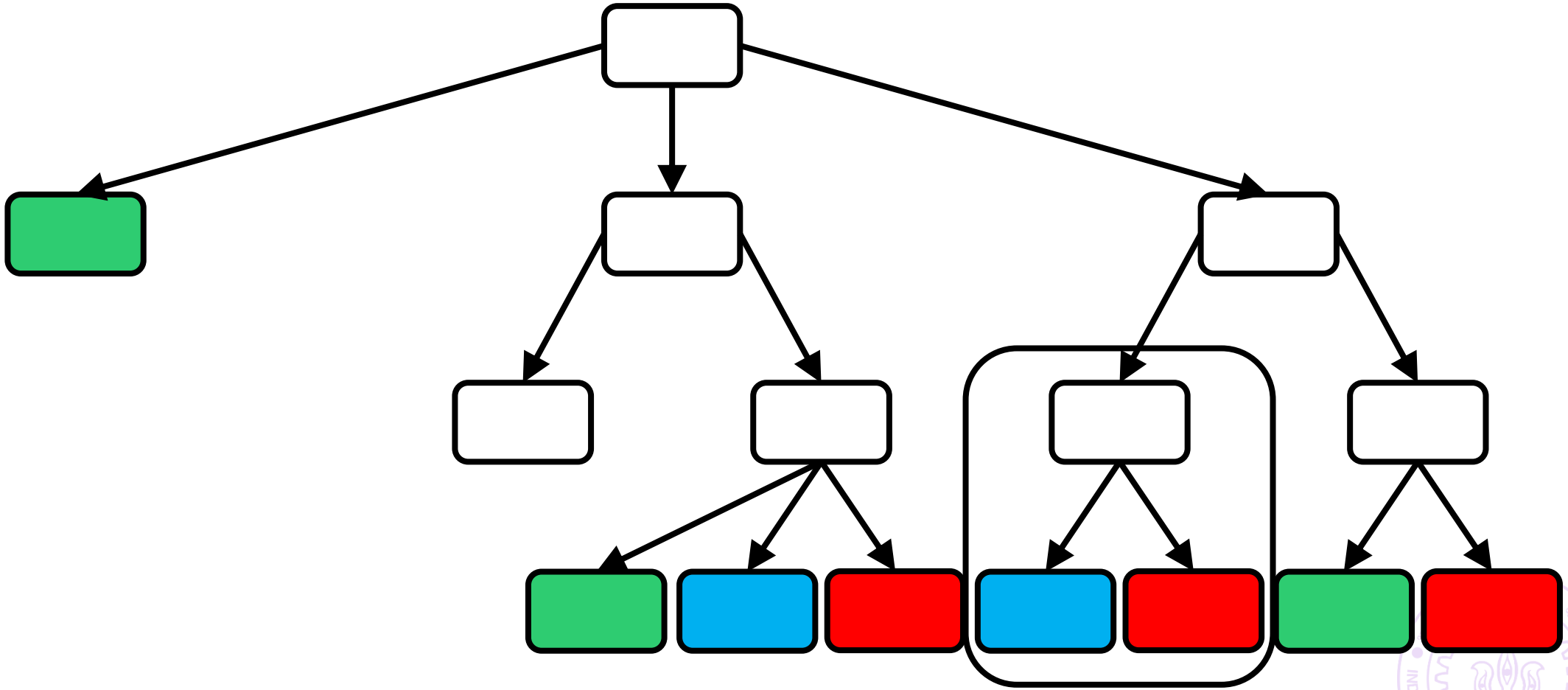
Decision Trees – all shapes and sizes



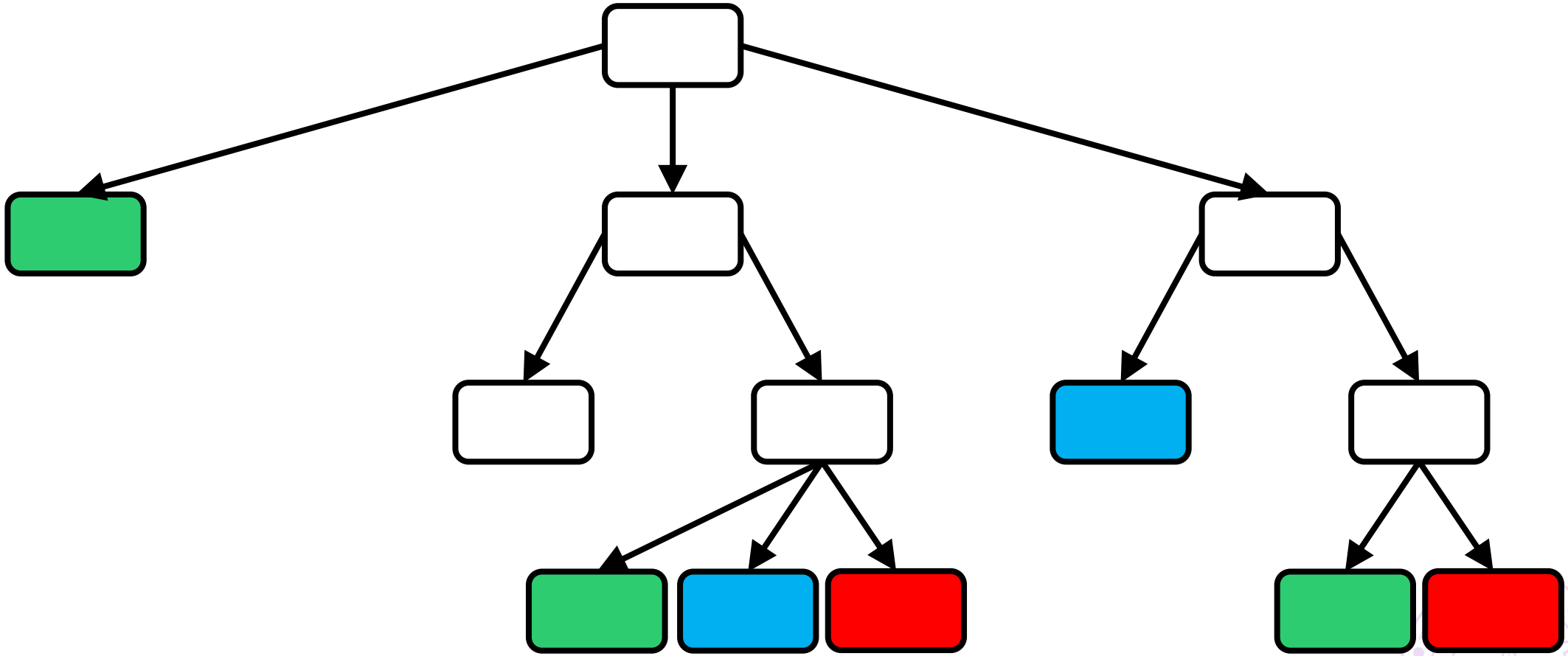
Decision Trees – all shapes and sizes



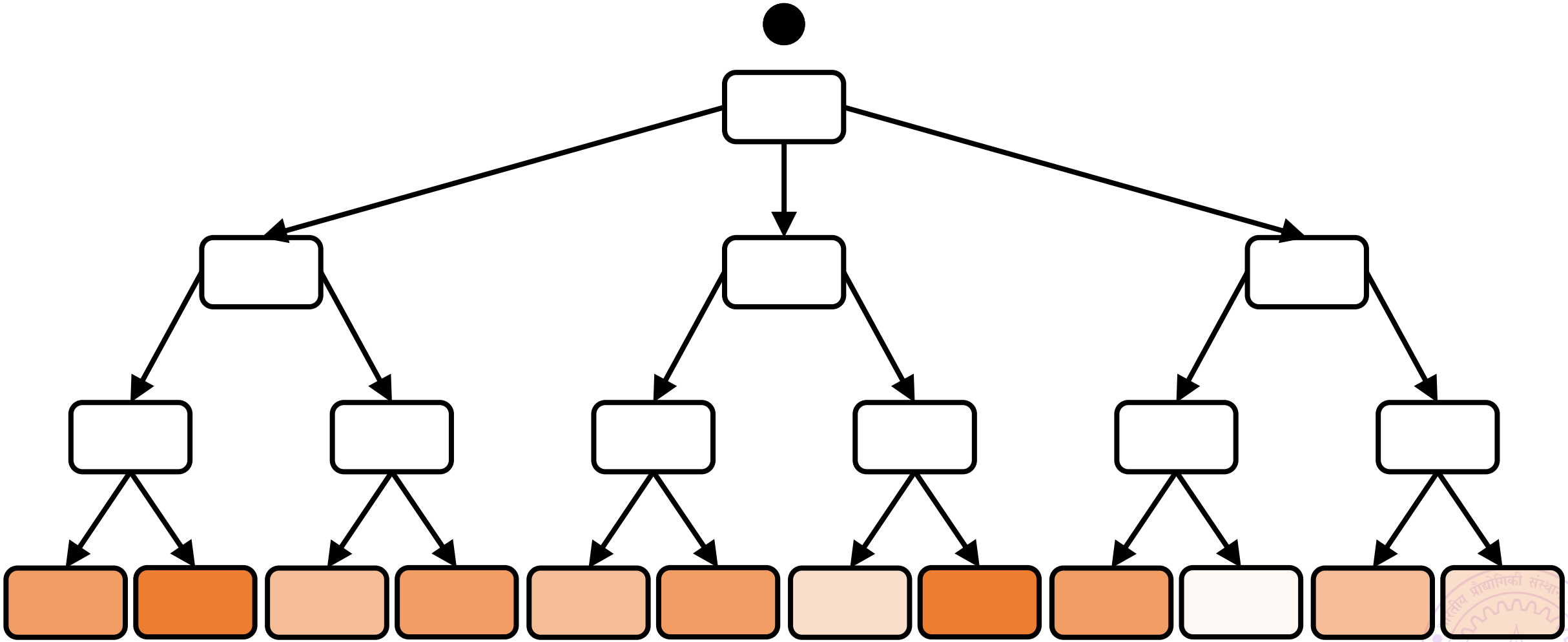
Decision Trees – all shapes and sizes



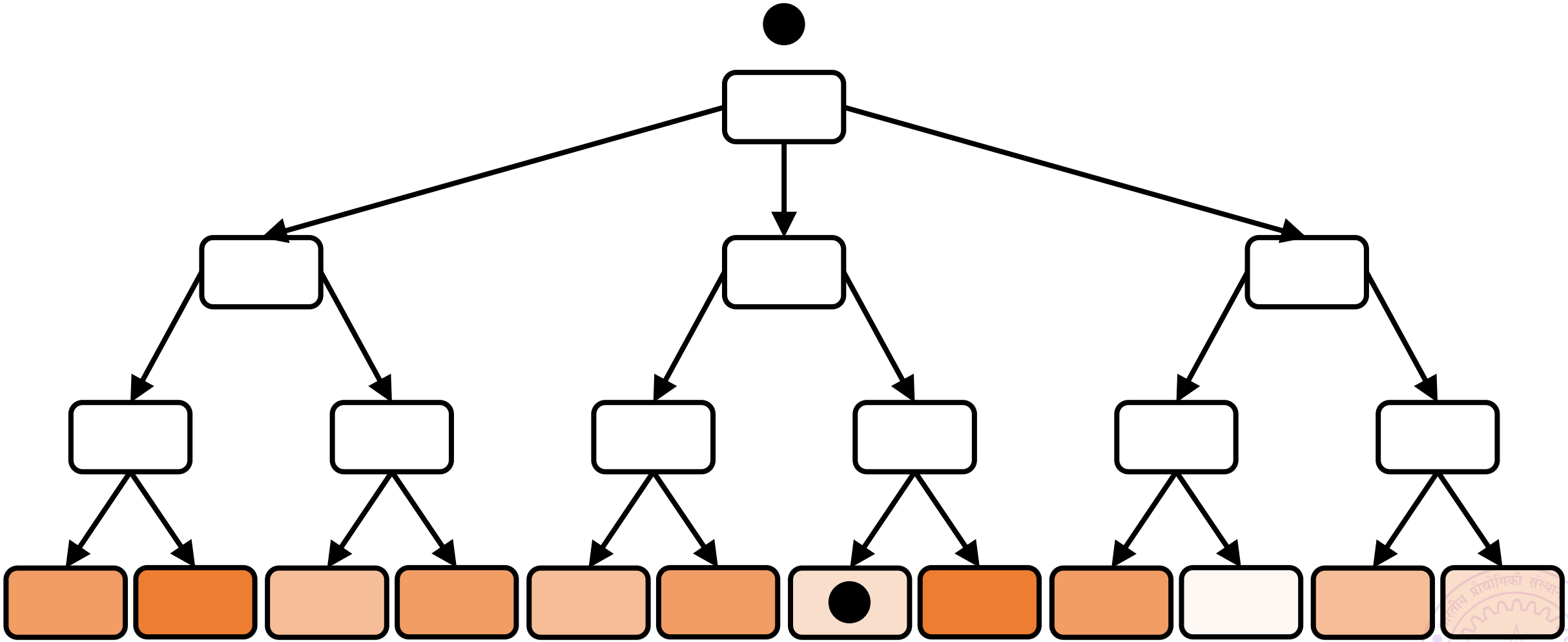
Decision Trees – all shapes and sizes



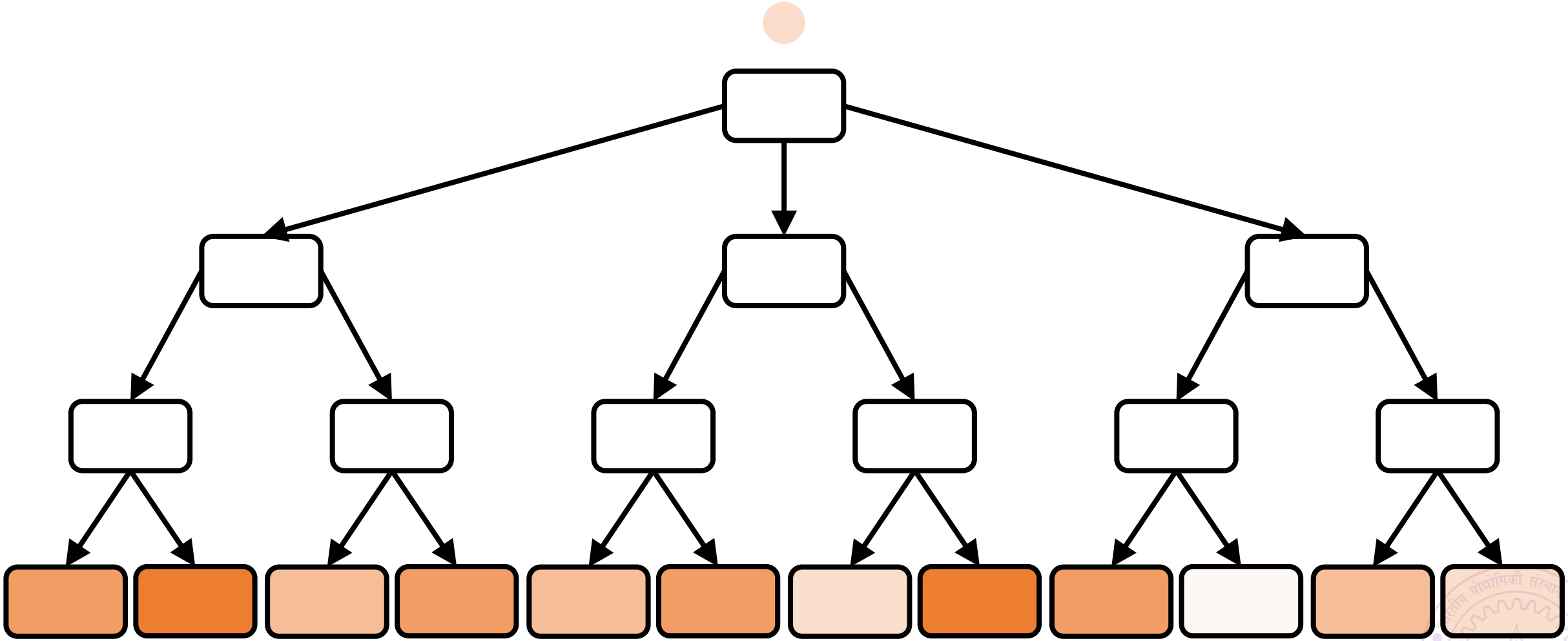
Regression with Decision Trees



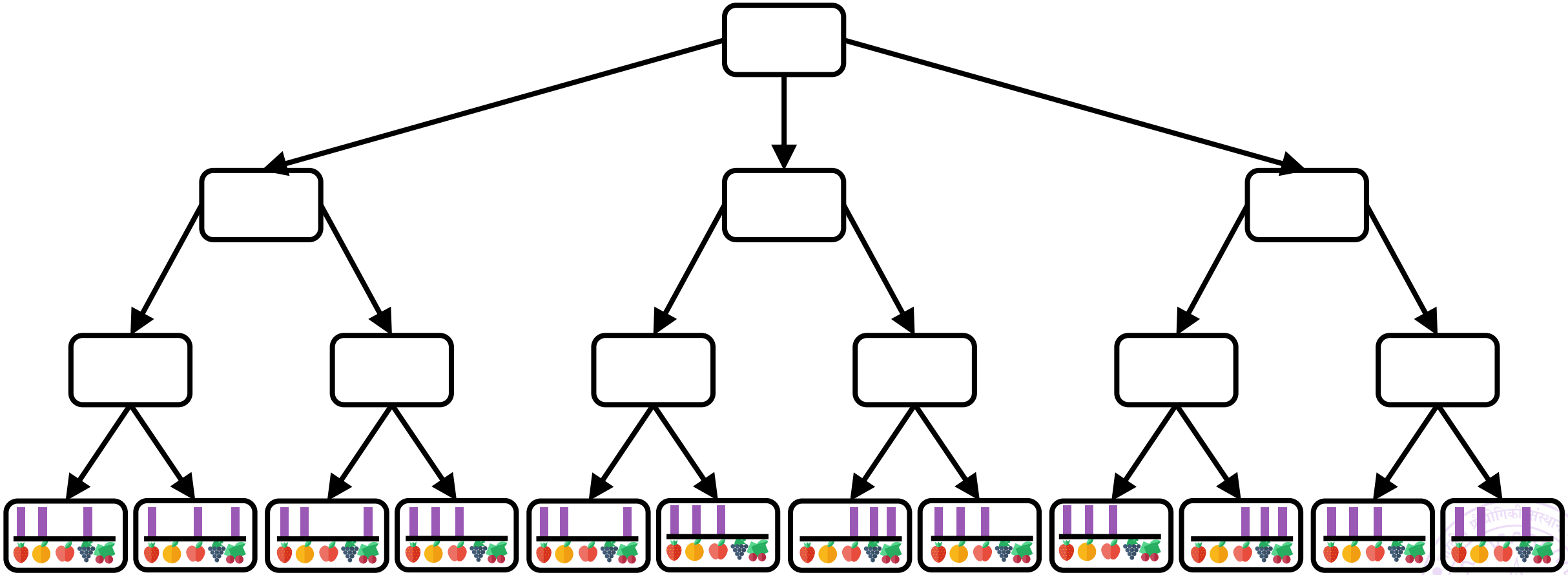
Regression with Decision Trees



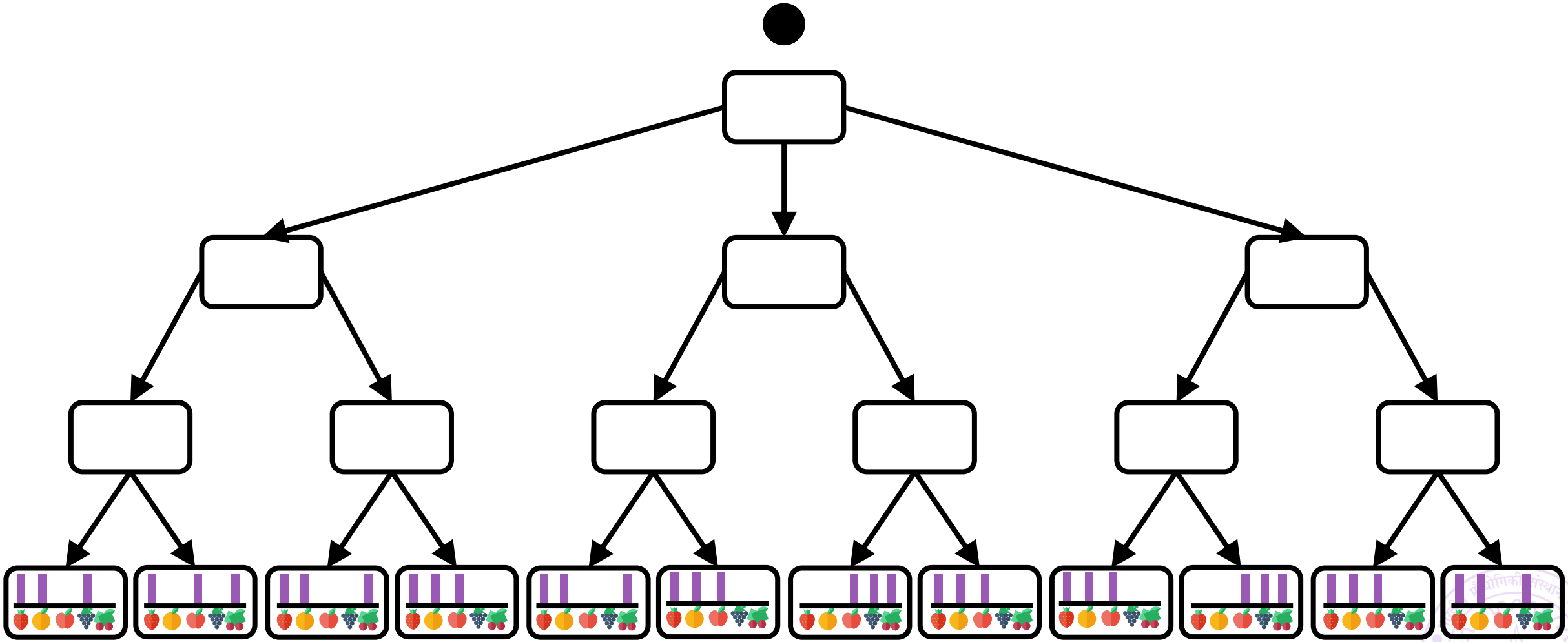
Regression with Decision Trees



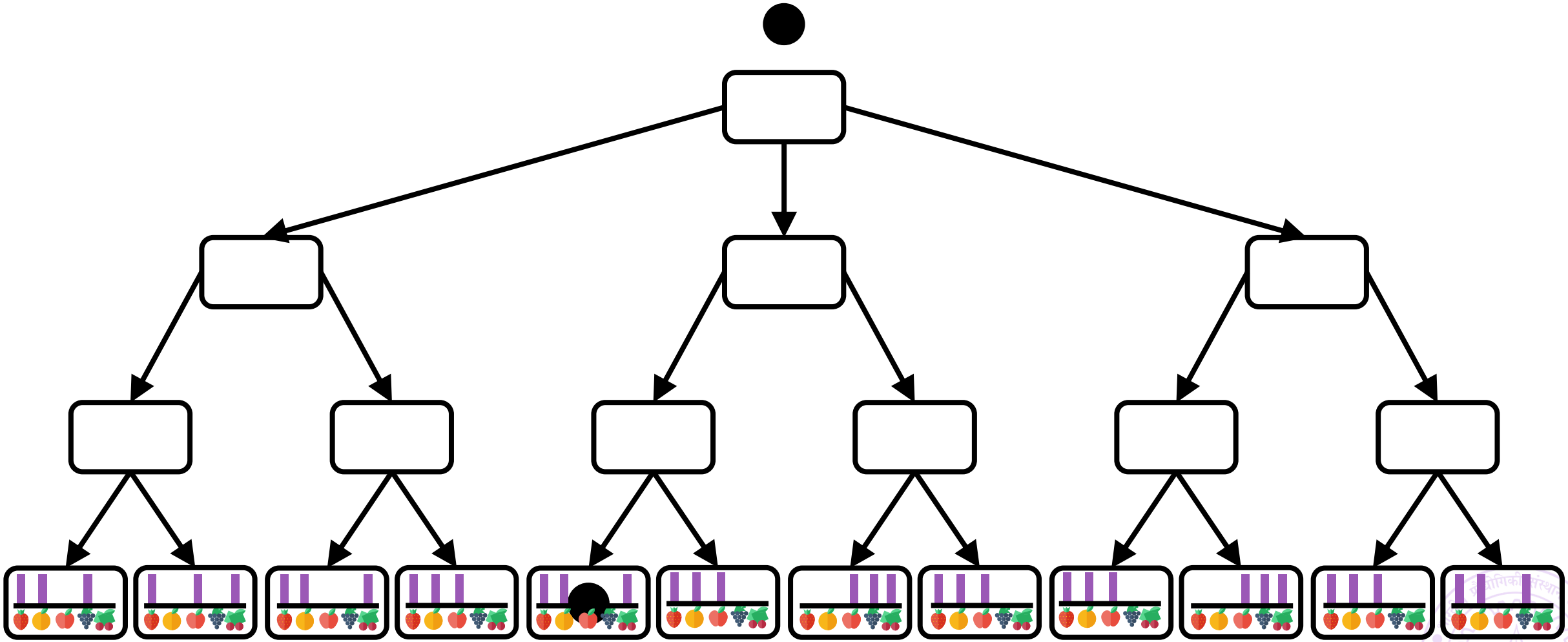
Multi-label Classification with Decision Trees



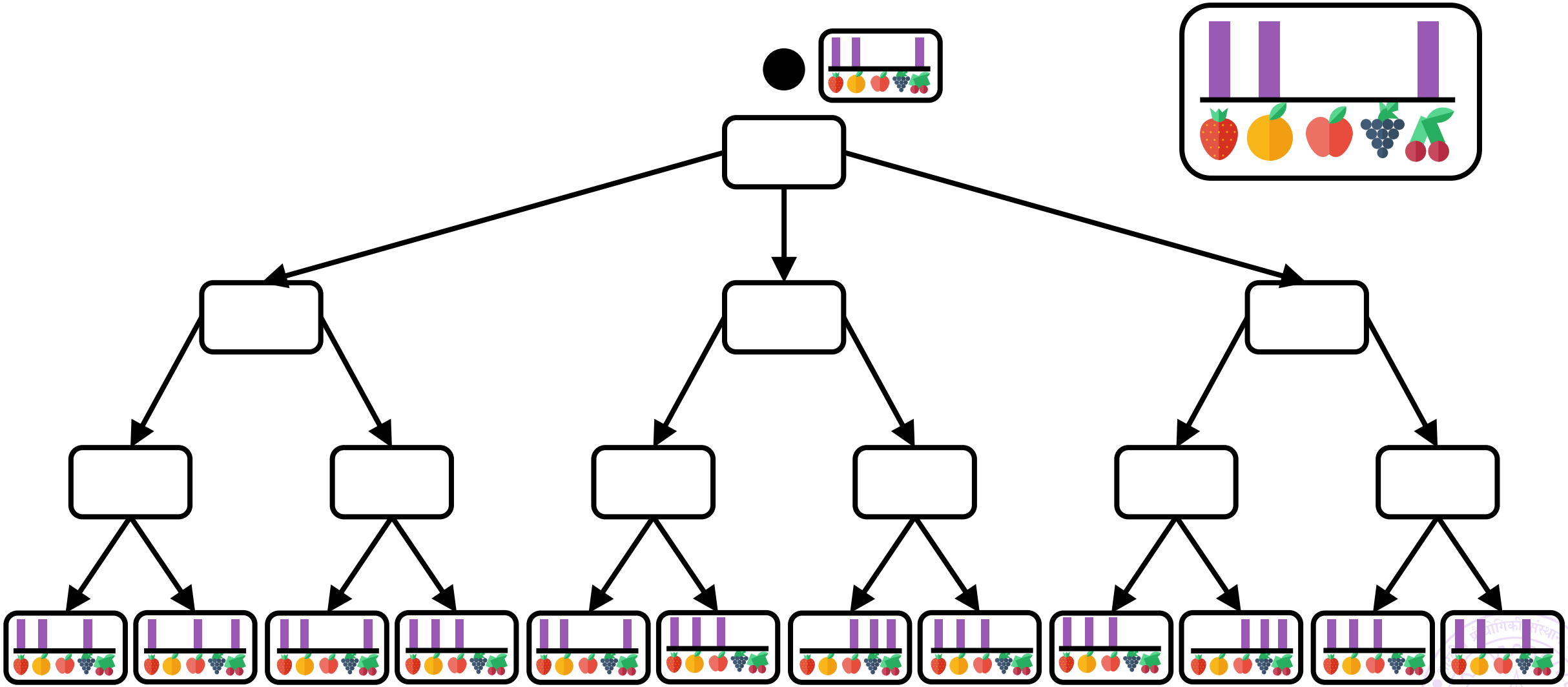
Multi-label Classification with Decision Trees



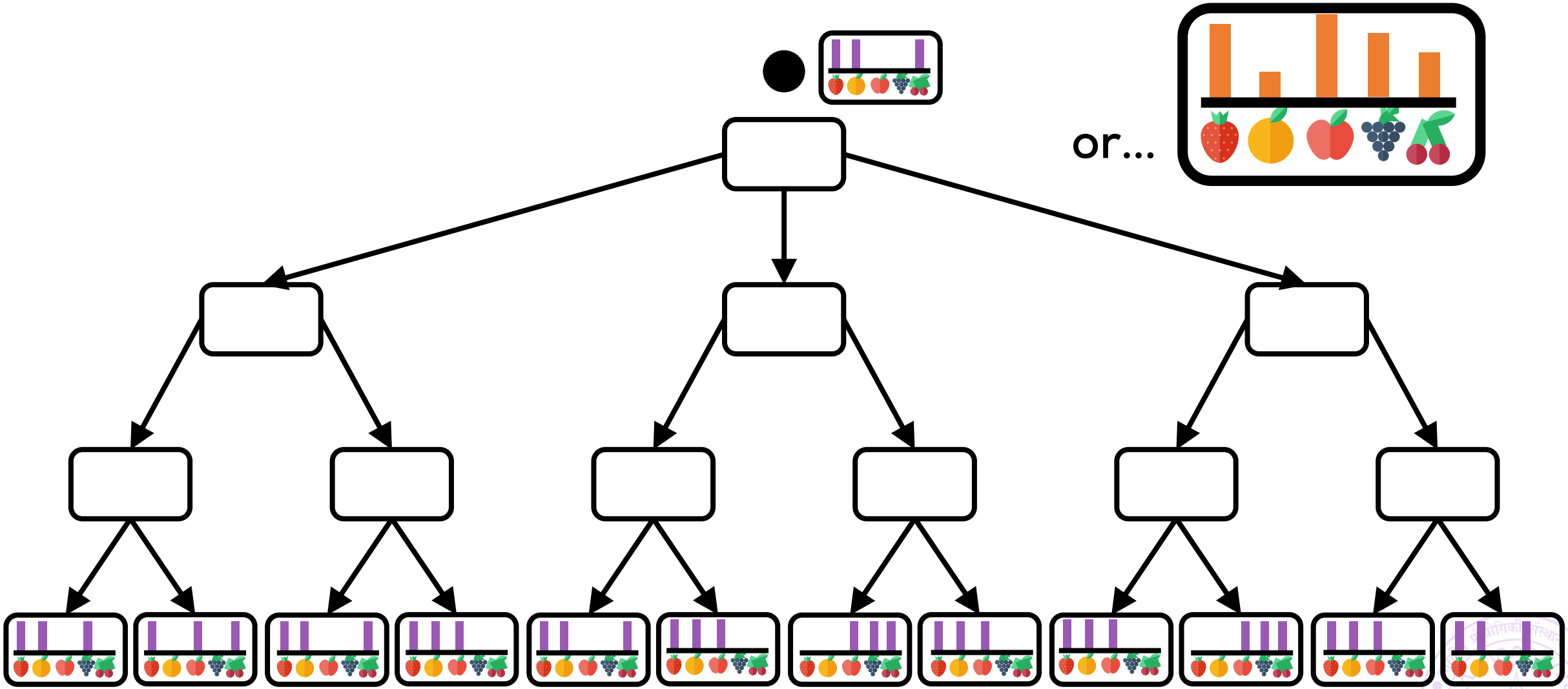
Multi-label Classification with Decision Trees



Multi-label Classification with Decision Trees



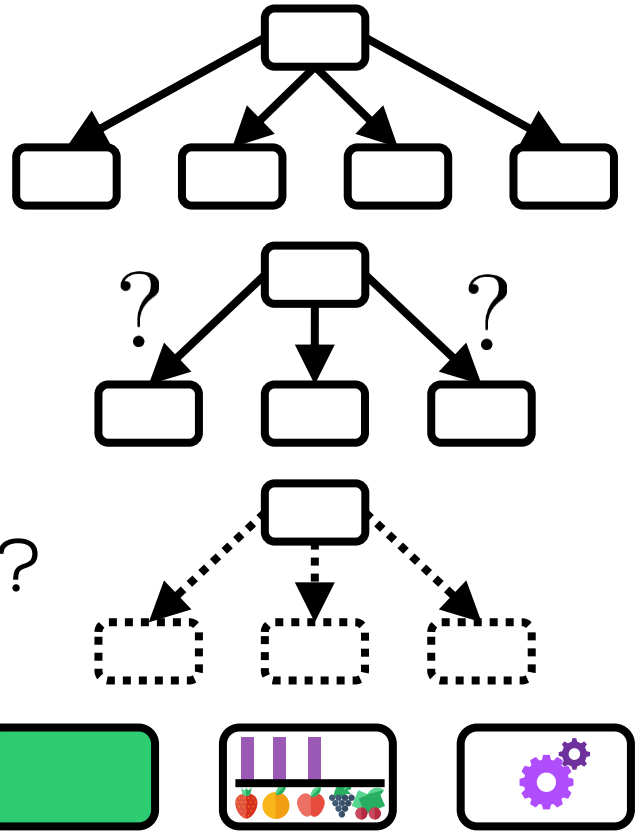
Multi-label Classification with Decision Trees

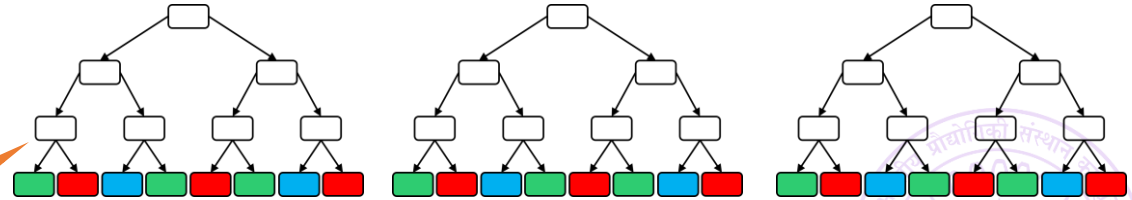


DT Learning Algorithms

The Questions that Matter

- How many children should a node have?
- How to send data points to children?
- When to stop splitting and make the node a leaf?
- What to do at a leaf?
- How many trees to train?

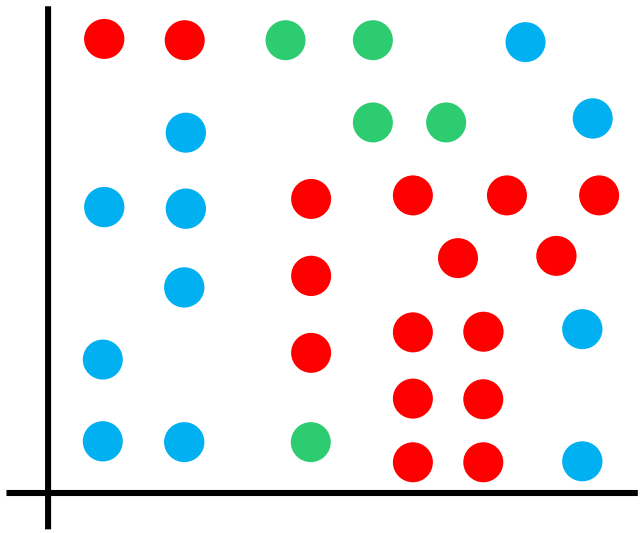


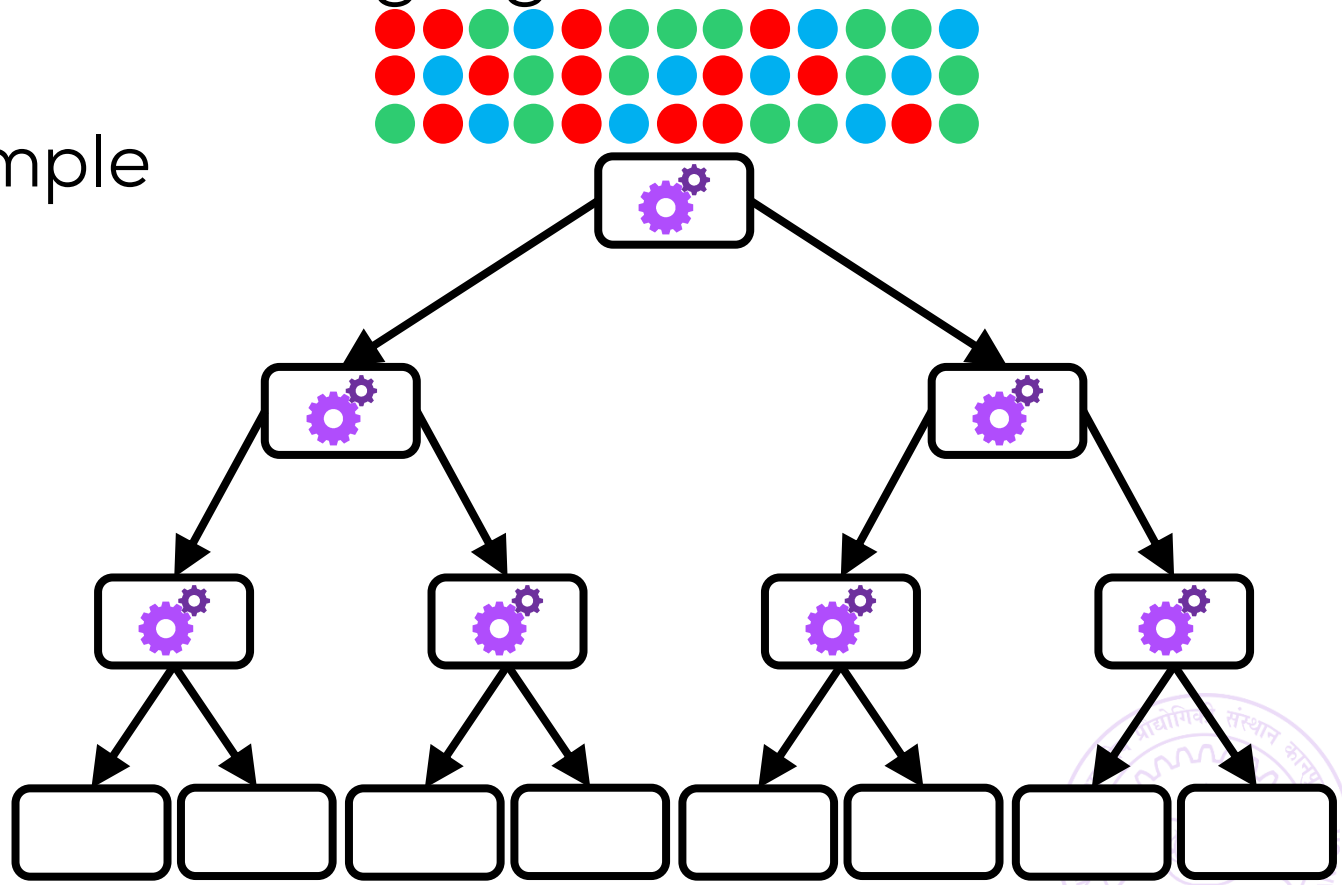


Model selection, memory constraints

Leaf Action

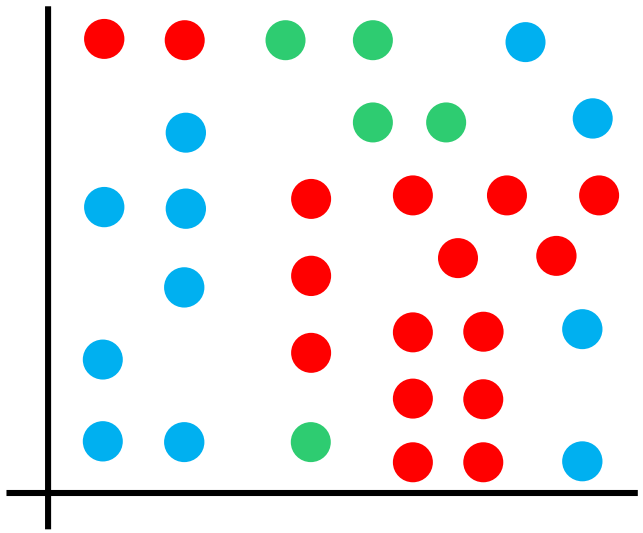
- Can take any (complicated) action at a leaf
  - Why not call another machine learning algorithm?
  - Useful trick to keep in mind
- For speed, keep leaf action simple
  - Simplest action – constant

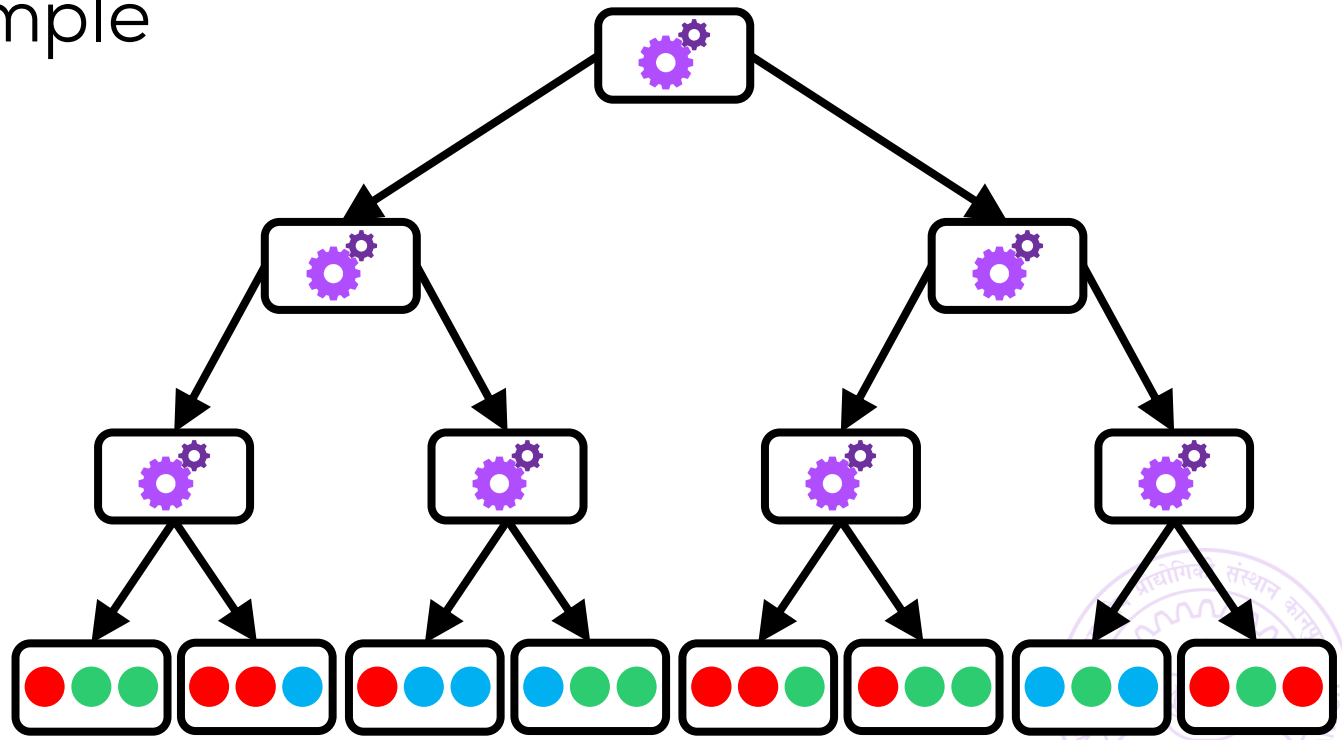




Leaf Action

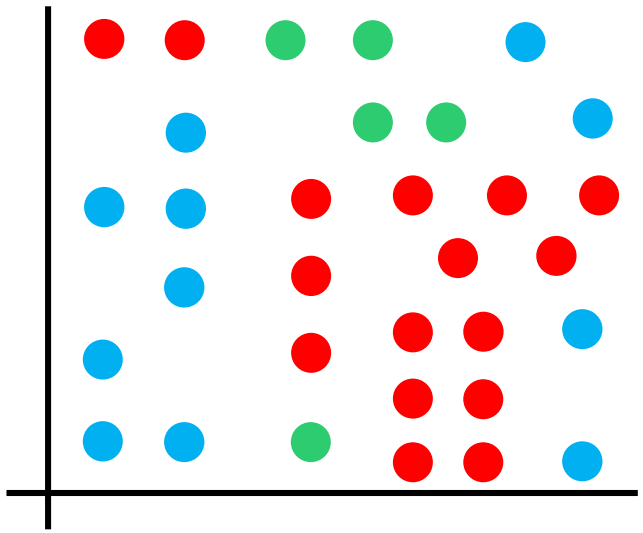
- Can take any (complicated) action at a leaf
  - Why not call another machine learning algorithm?
  - Useful trick to keep in mind
- For speed, keep leaf action simple
  - Simplest action – constant

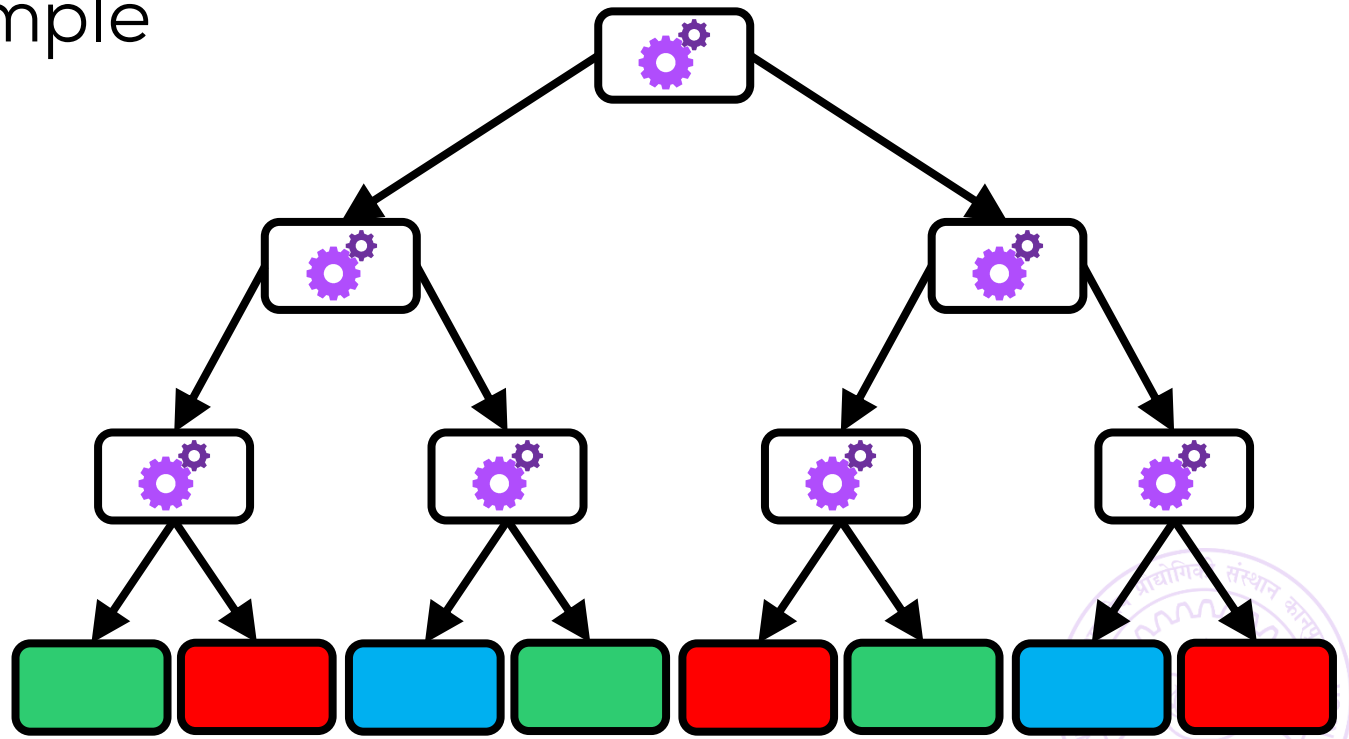




Leaf Action

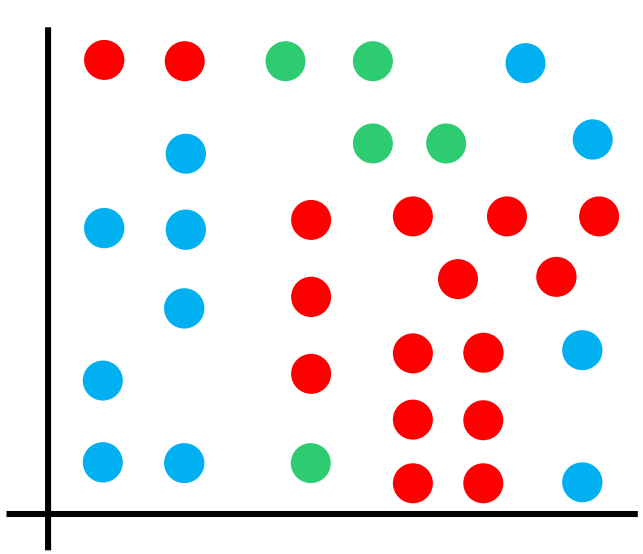
- Can take any (complicated) action at a leaf
  - Why not call another machine learning algorithm?
  - Useful trick to keep in mind
- For speed, keep leaf action simple
  - Simplest action – constant

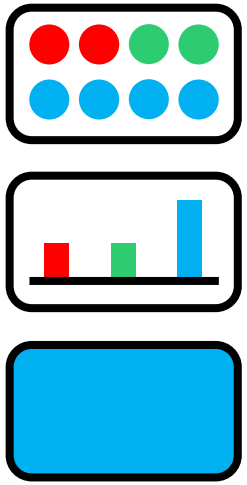


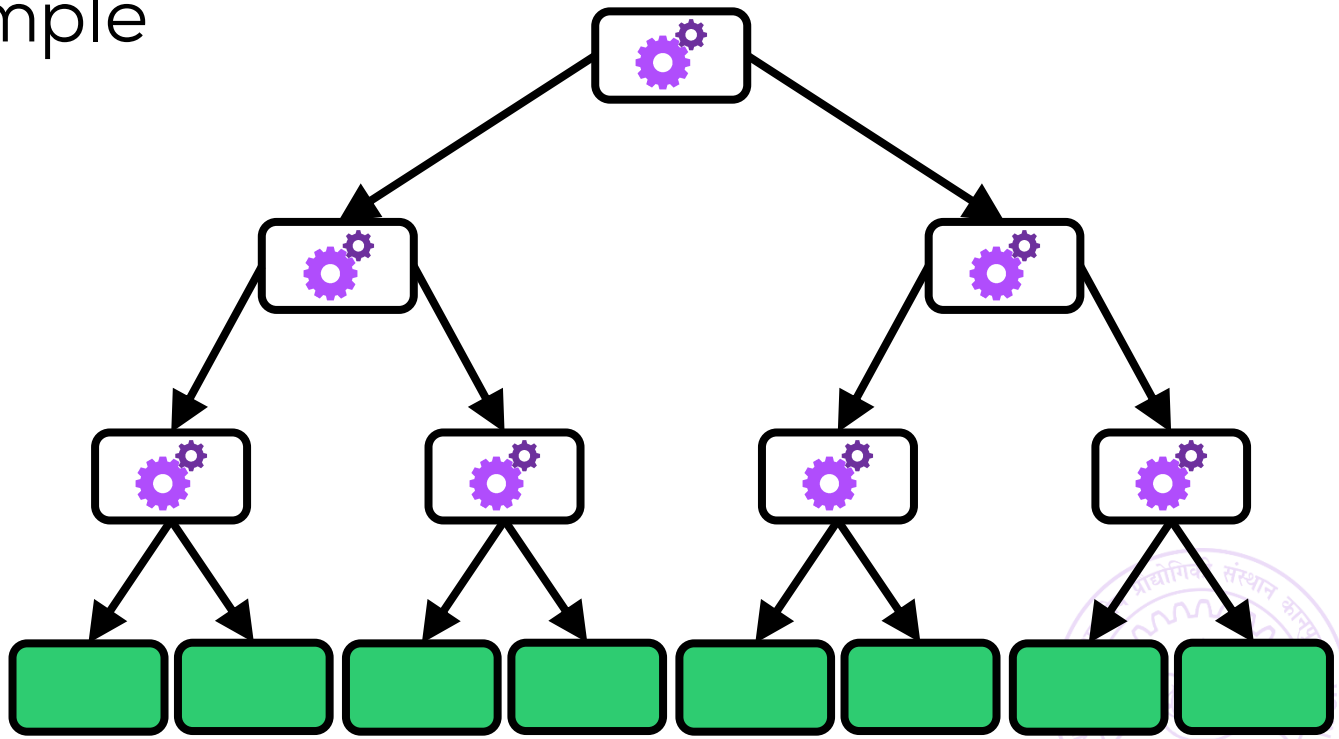


Leaf Action

- Can take any (complicated) action at a leaf
  - Why not call another machine learning algorithm?
  - Useful trick to keep in mind
- For speed, keep leaf action simple
  - Simplest action – constant

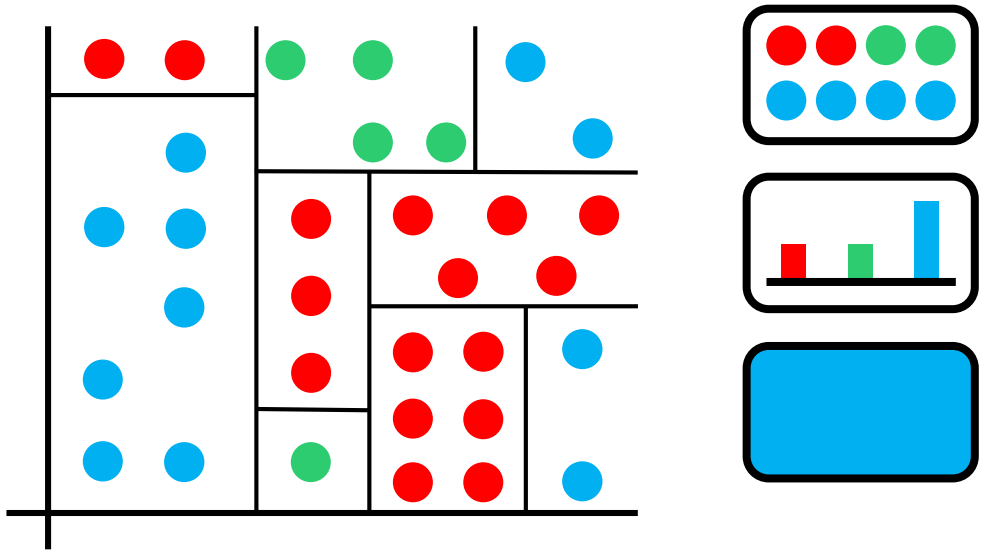


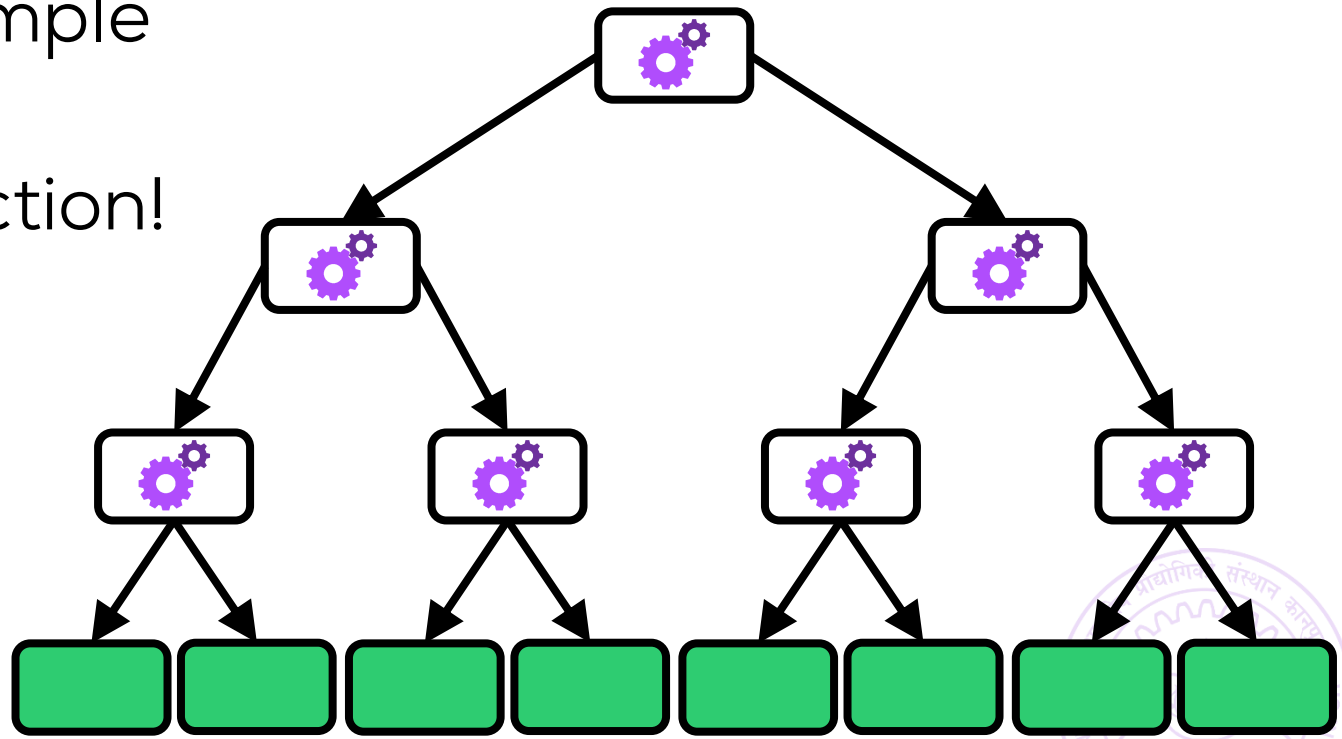




Leaf Action

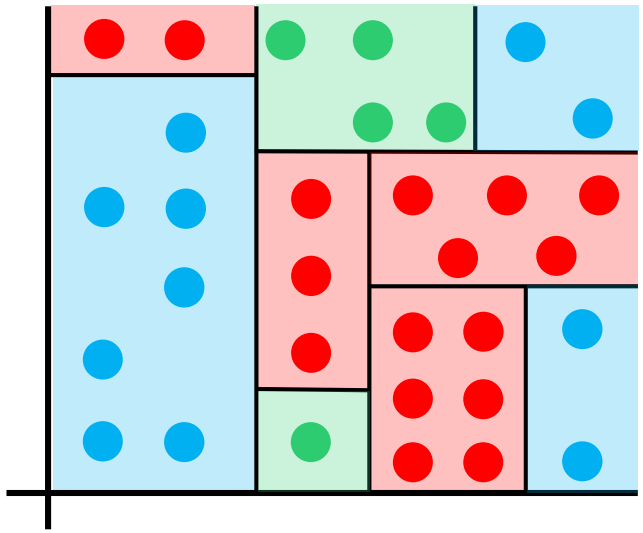
- Can take any (complicated) action at a leaf
  - Why not call another machine learning algorithm?
  - Useful trick to keep in mind
- For speed, keep leaf action simple
  - Simplest action – constant
  - DT piecewise constant function!

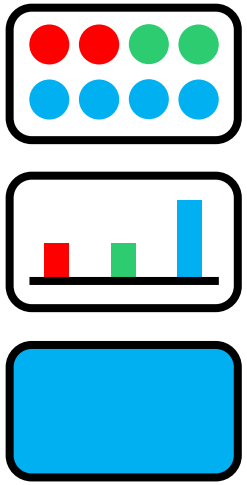


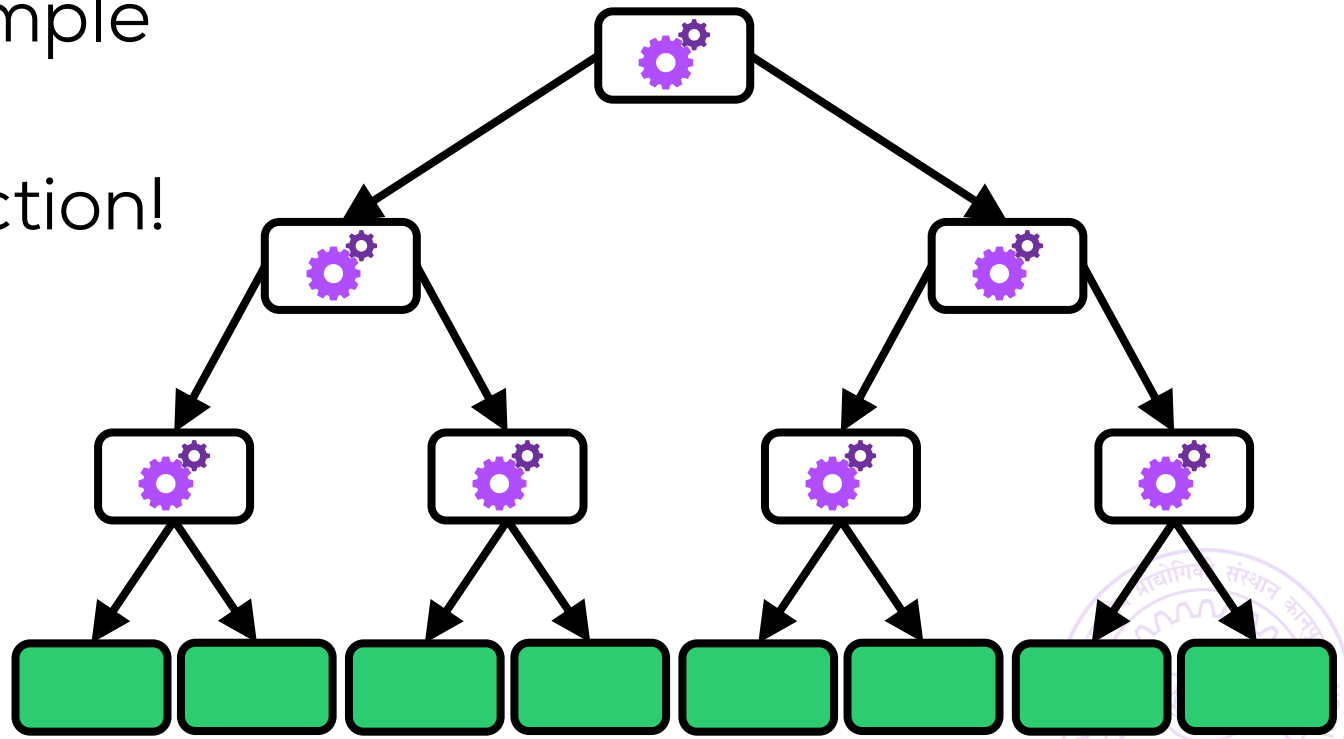


Leaf Action

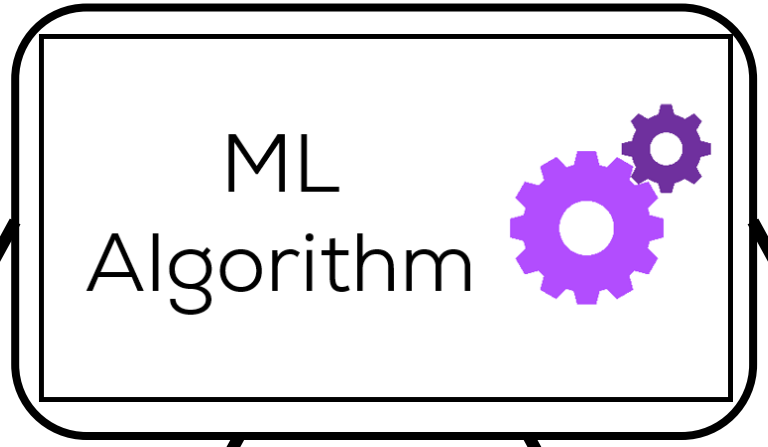
- Can take any (complicated) action at a leaf
  - Why not call another machine learning algorithm?
  - Useful trick to keep in mind
- For speed, keep leaf action simple
  - Simplest action – constant
  - DT piecewise constant function!







Node Splitting



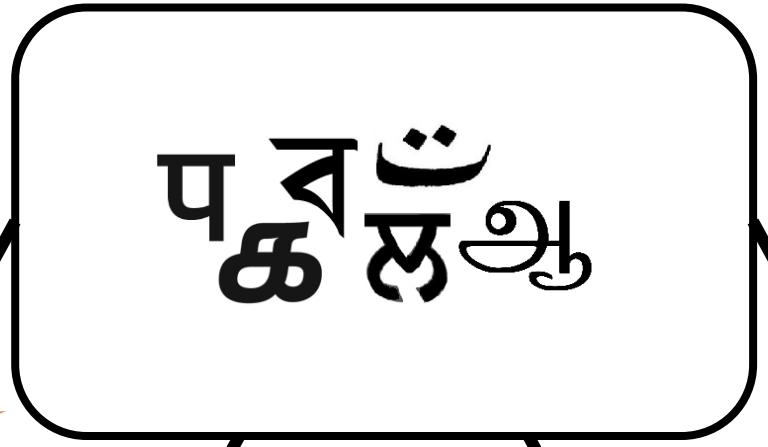
A "classification" problem in itself!

Usually a simple "stump"

Based on only one feature: speed!

One child per value feature can take!

Node Splitting



A "classification" problem in itself!

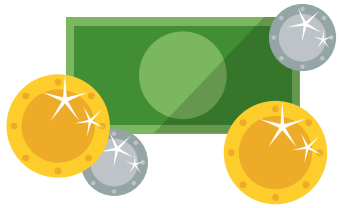
Discrete features easy to handle

Usually a simple "stump"

Based on only one feature: speed!

One child per value feature can take!

Node Splitting



Low, Medium, High

A "classification" problem in itself!

Discrete features easy to handle

Usually a simple "stump"

Based on only one feature: speed!

One child per value feature can take!

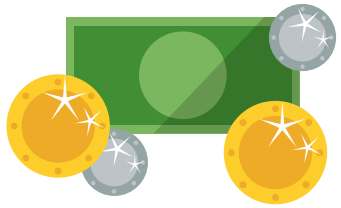
How finely?

Continuous features discretized

Node Splitting

Do "multi-label" splits make sense?

A "classification" problem in itself!



Low, Medium, High

Discrete features easy to handle

Usually a simple "stump"

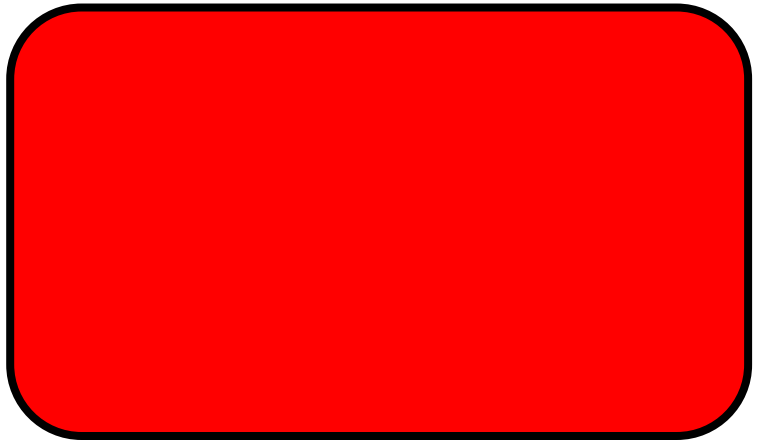
Based on only one feature: speed!

One child per value feature can take!

How finely?

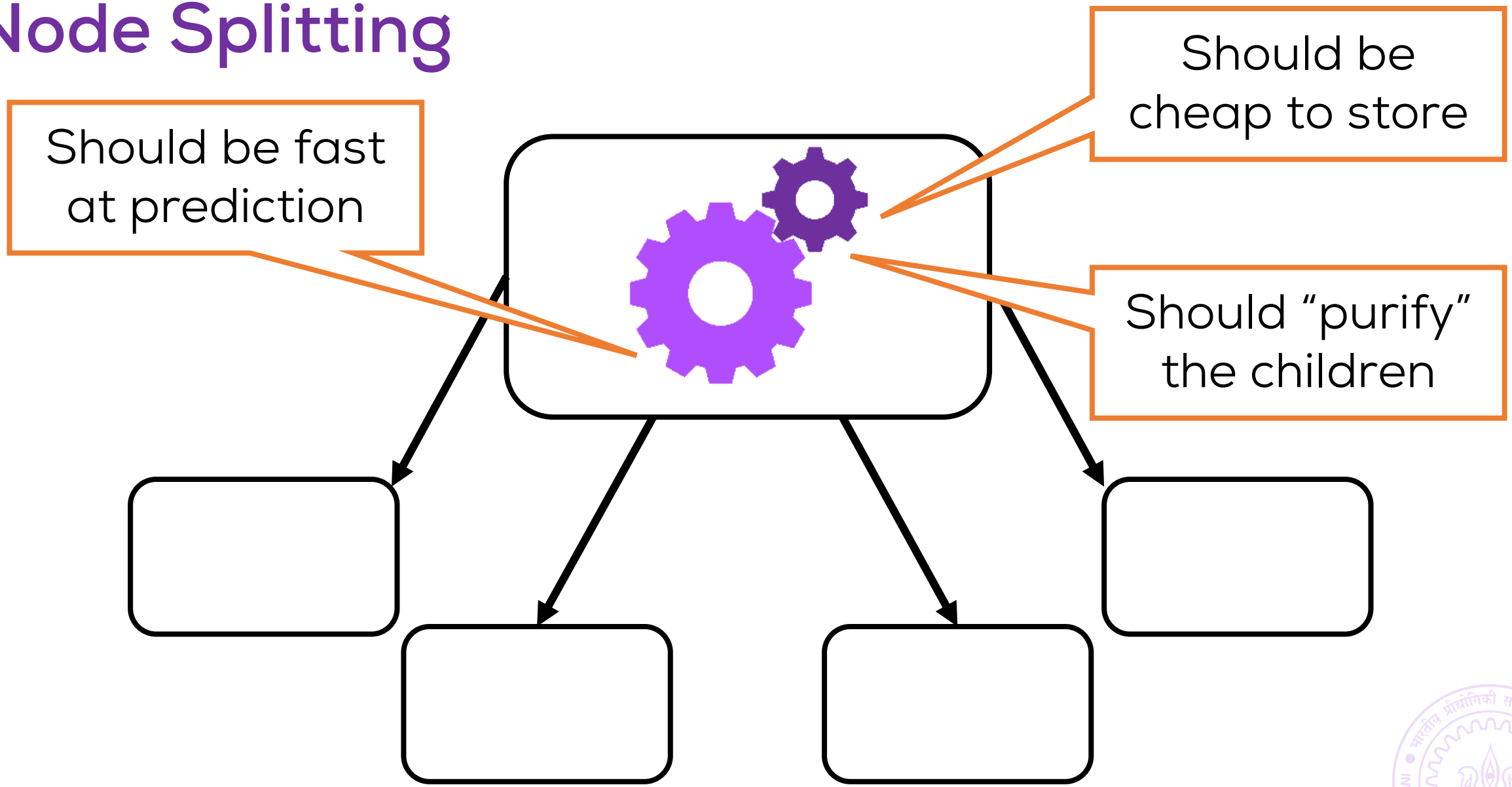
Continuous features discretized

Node Splitting

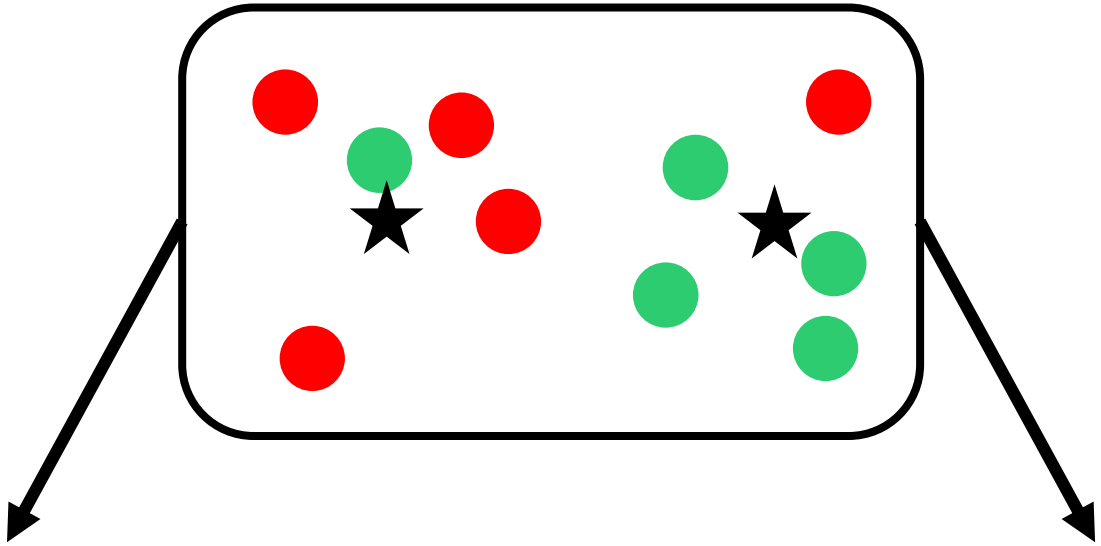


A pure node needs no children

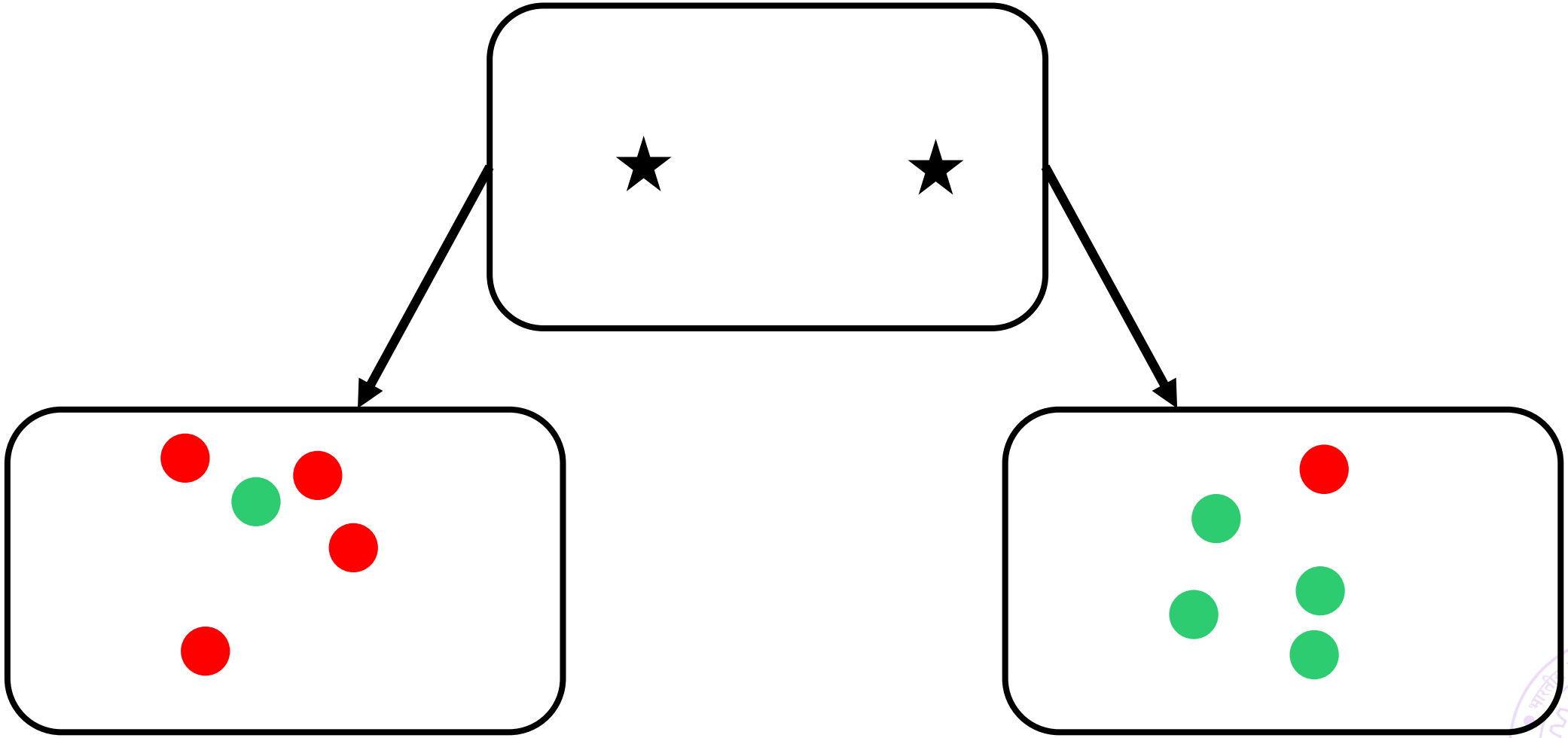
Node Splitting



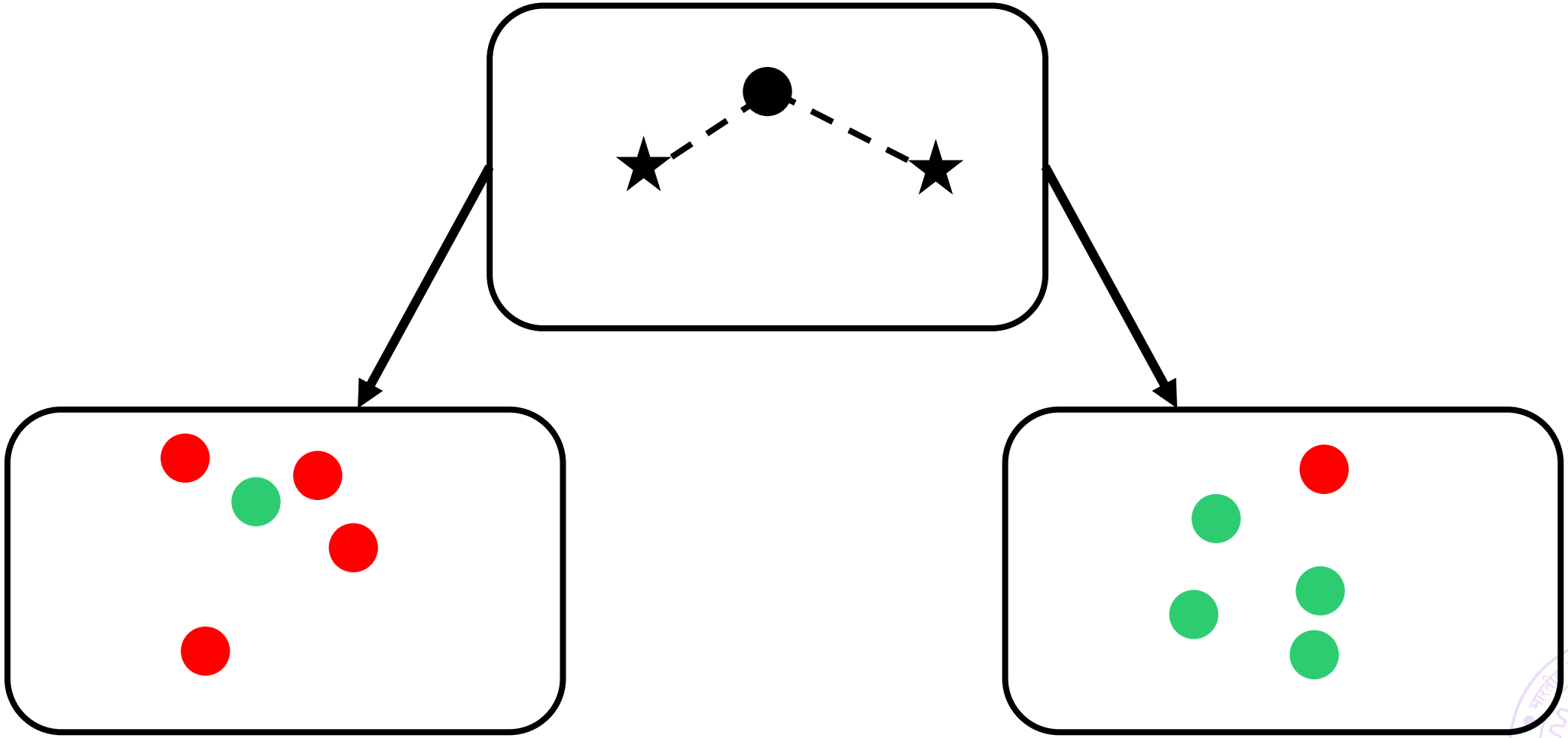
Node Splitting via prototypes (clustering)



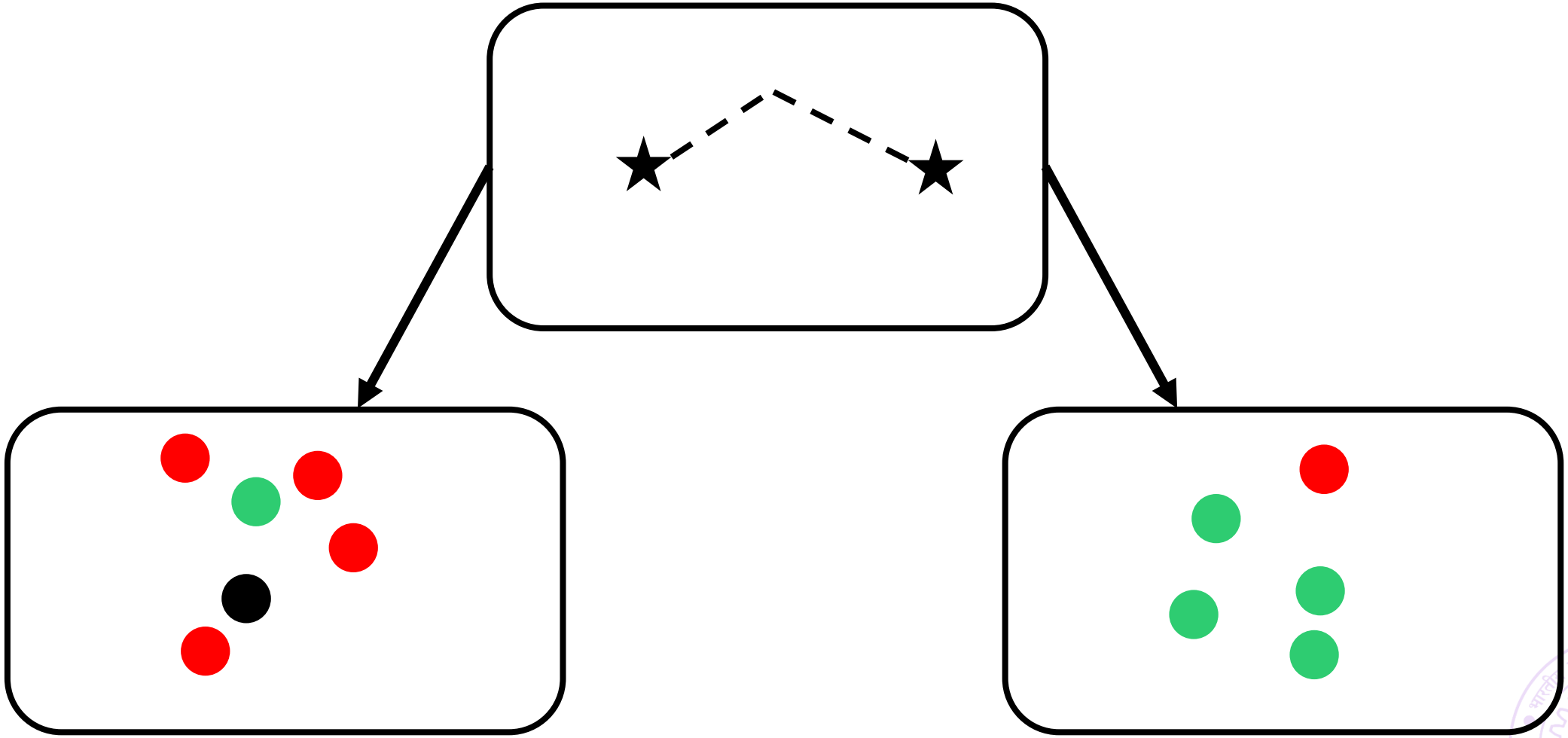
Node Splitting via prototypes (clustering)



Node Splitting via prototypes (clustering)

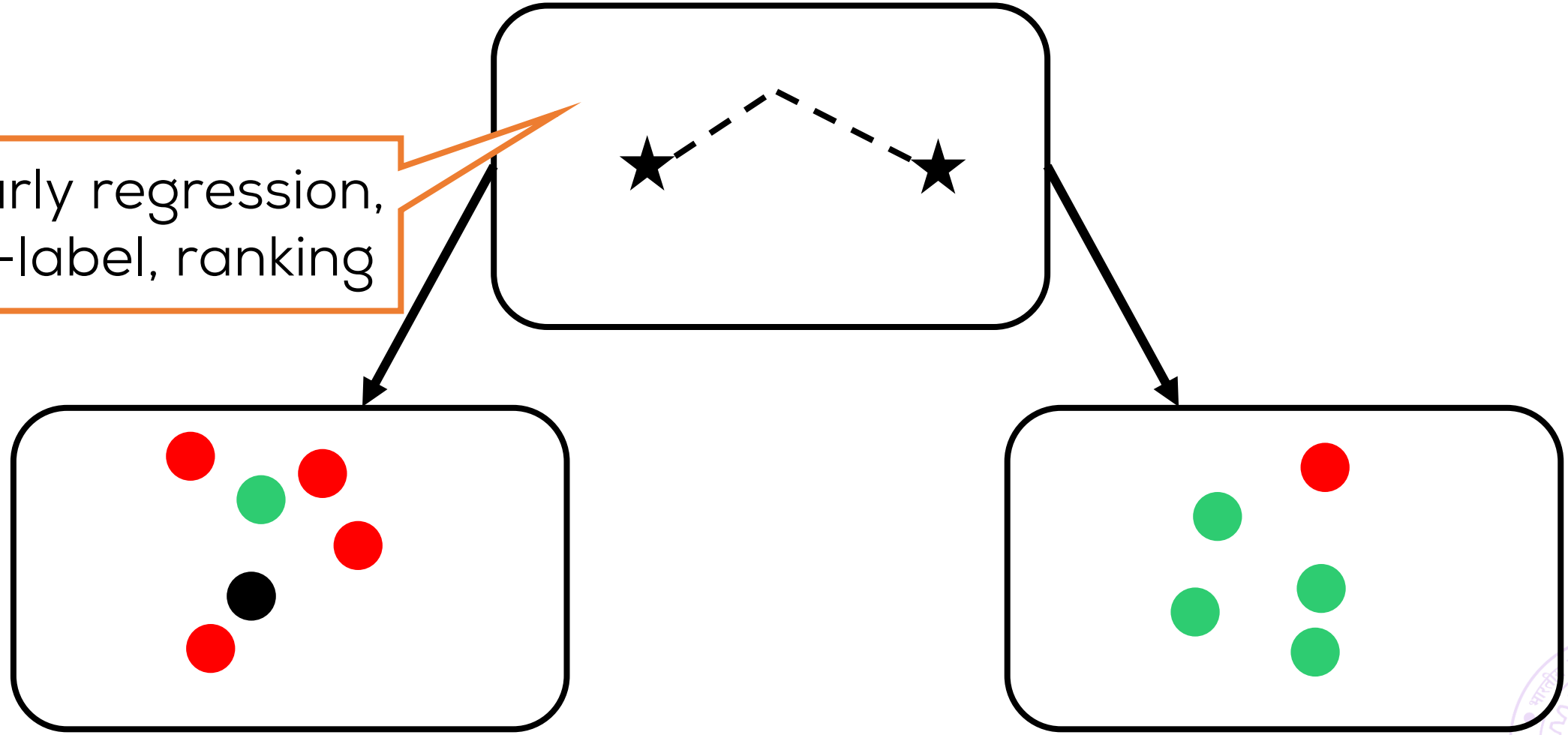


Node Splitting via prototypes (clustering)

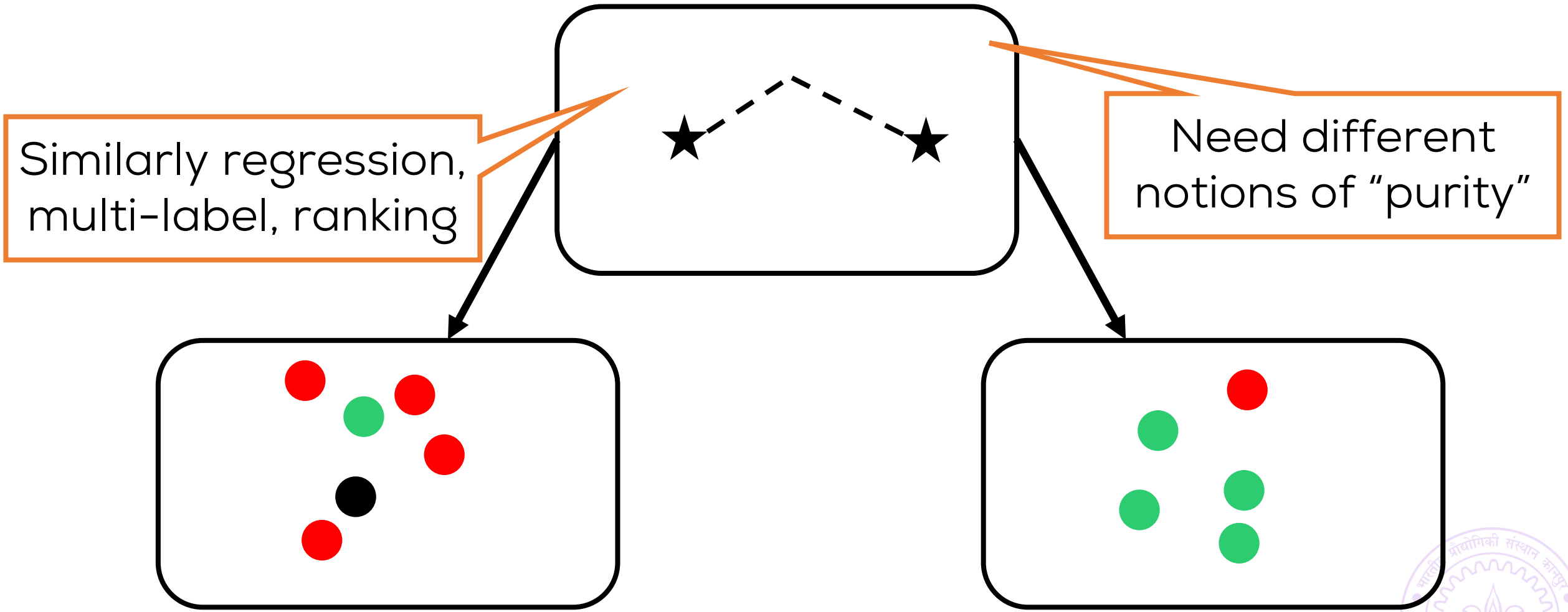


Node Splitting via prototypes (clustering)

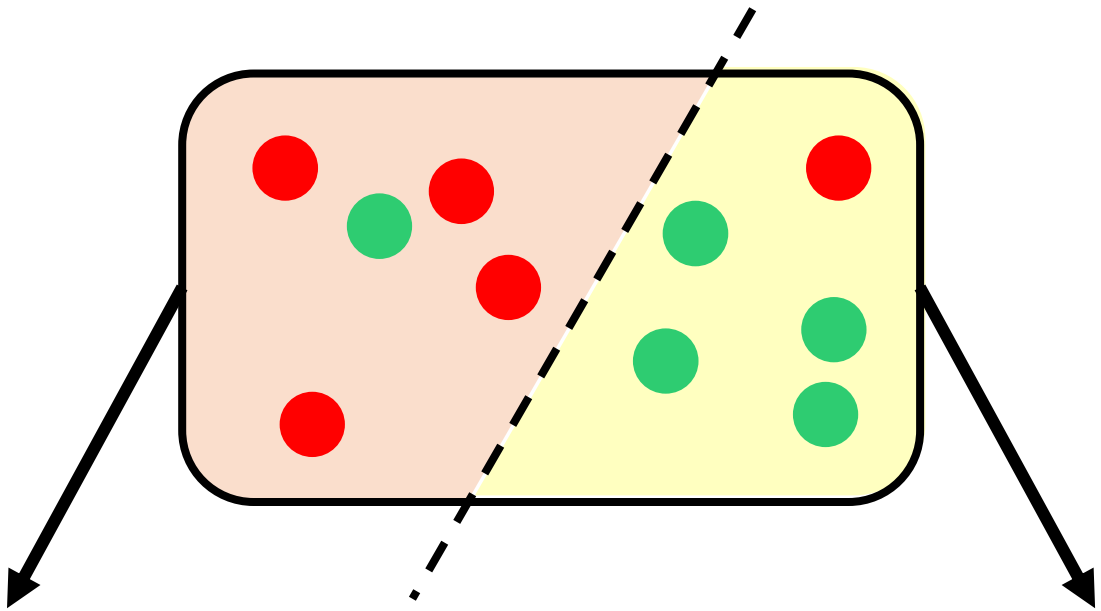
Similarly regression, multi-label, ranking



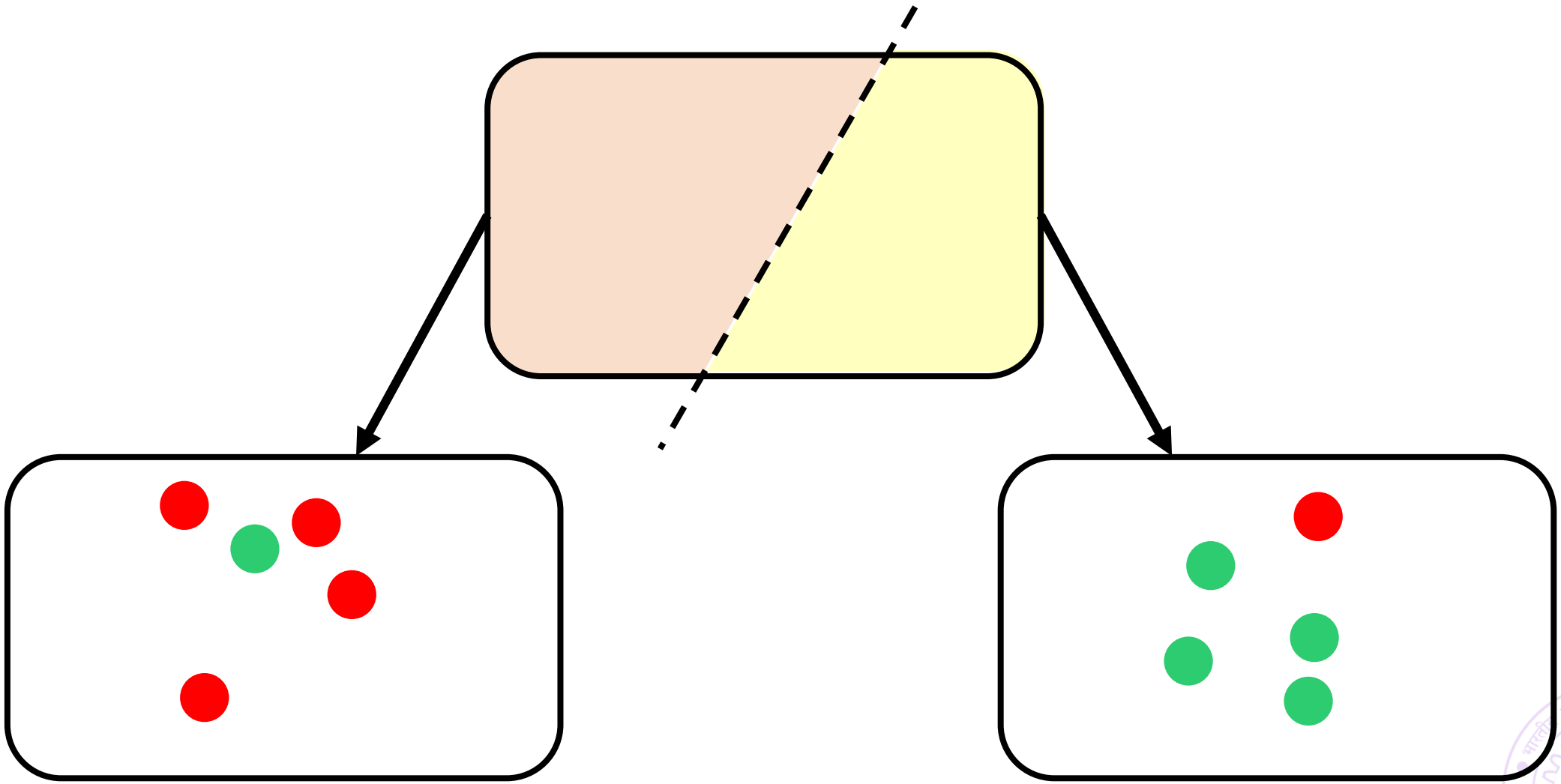
Node Splitting via prototypes (clustering)



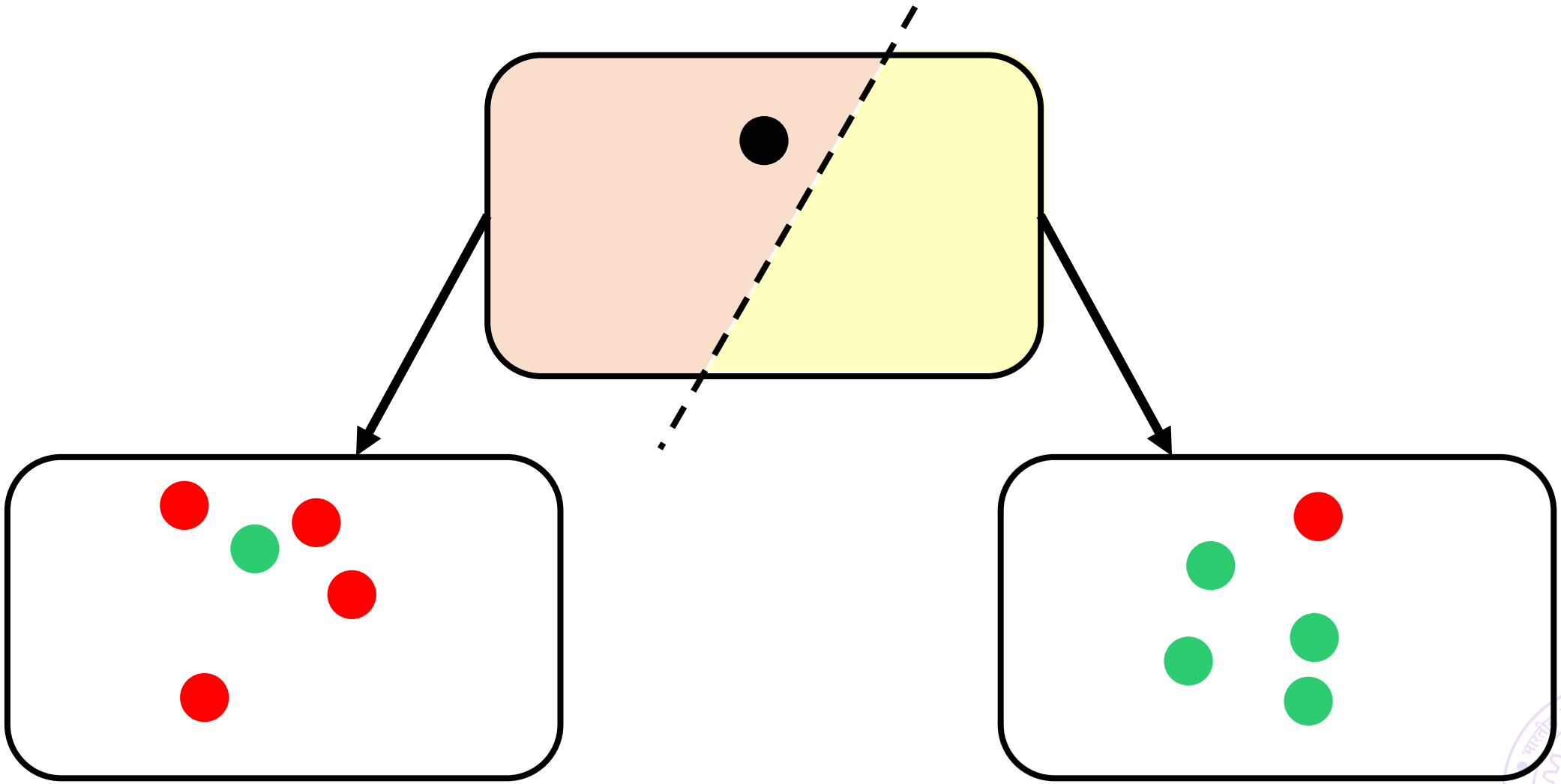
Node Splitting via linear classifiers



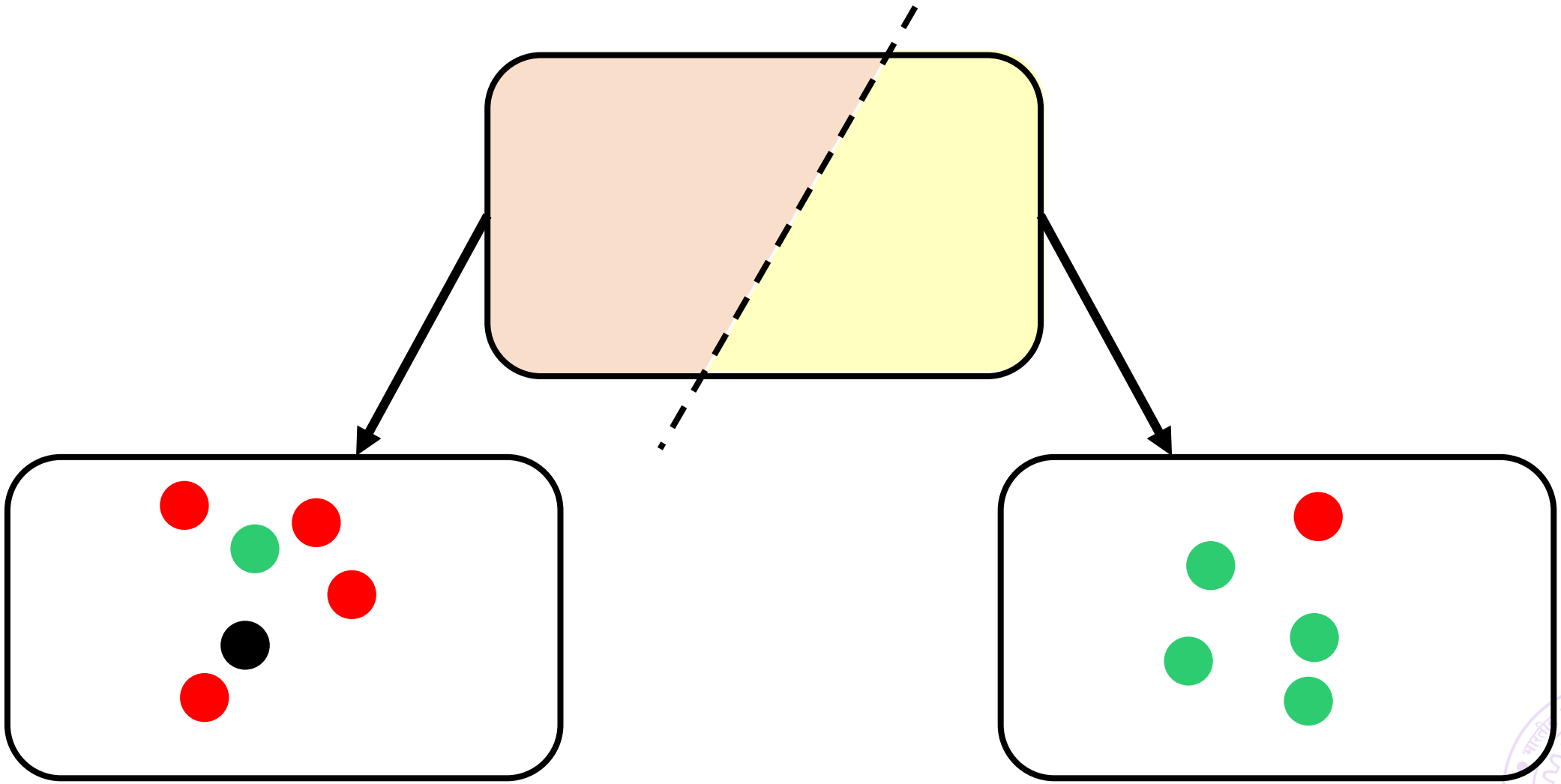
Node Splitting via linear classifiers



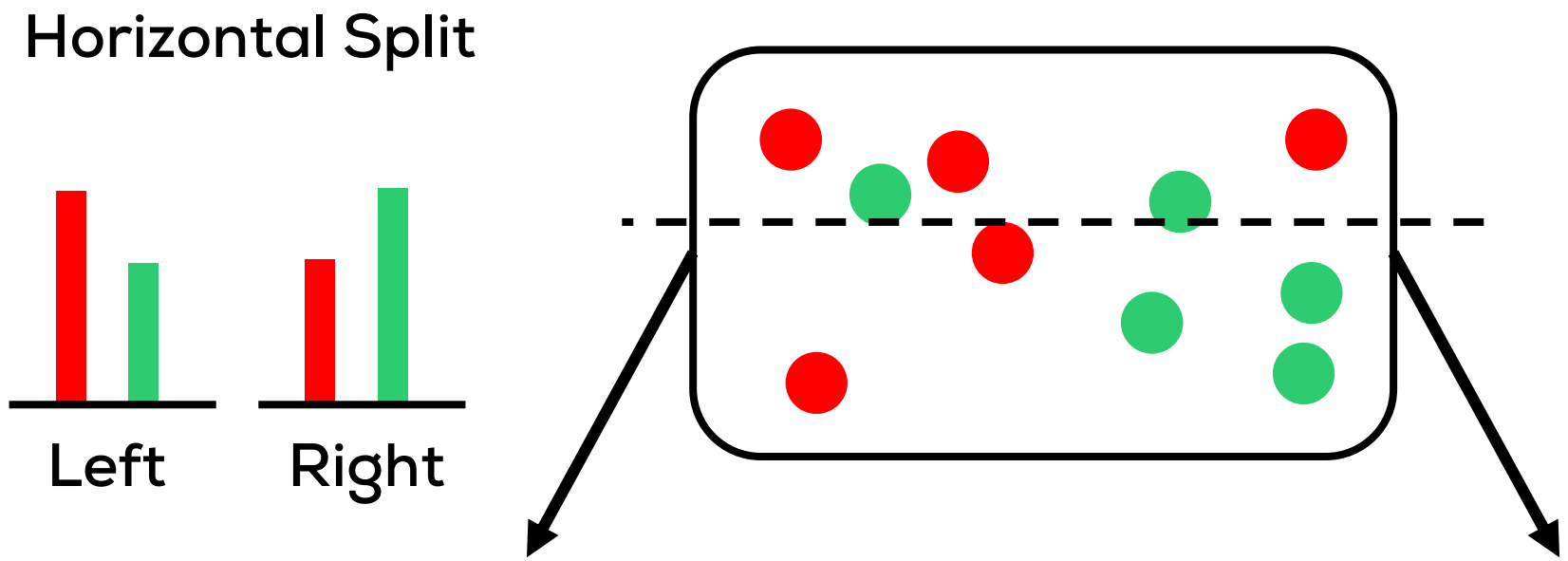
Node Splitting via linear classifiers



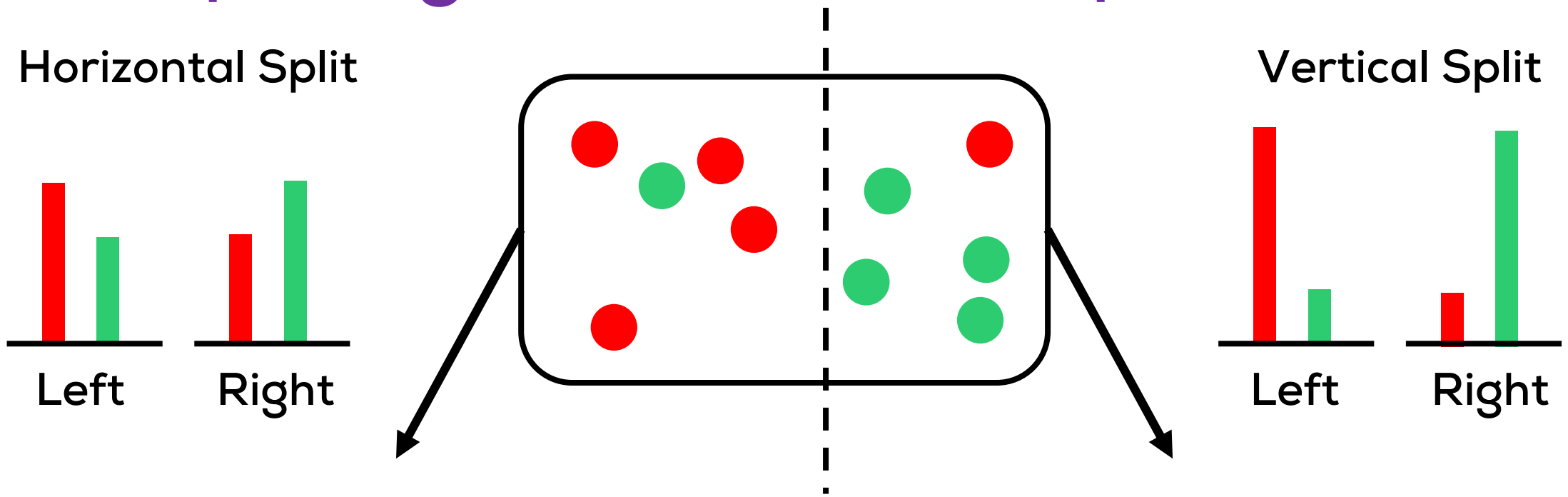
Node Splitting via linear classifiers



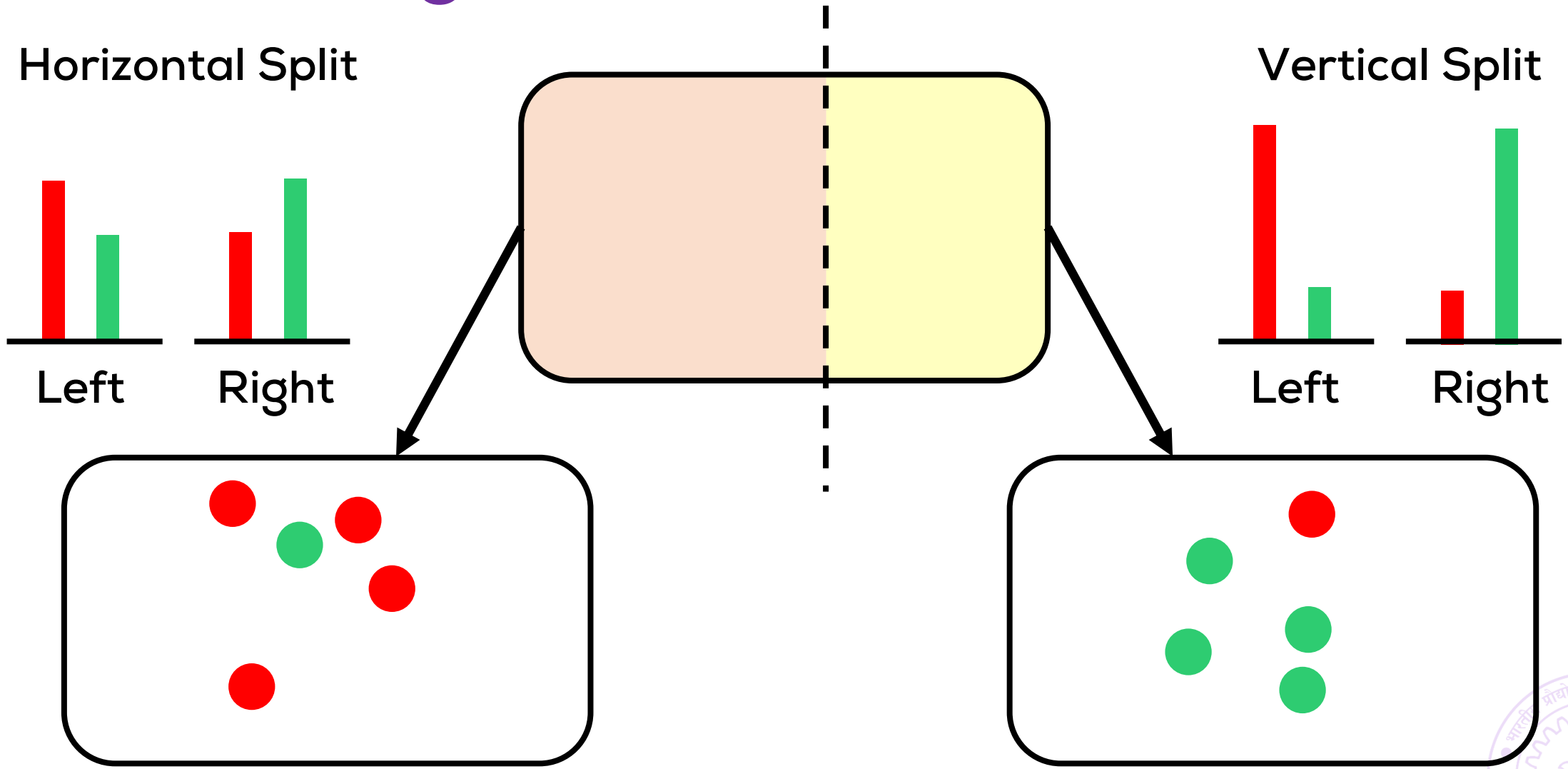
Node Splitting via feature stumps



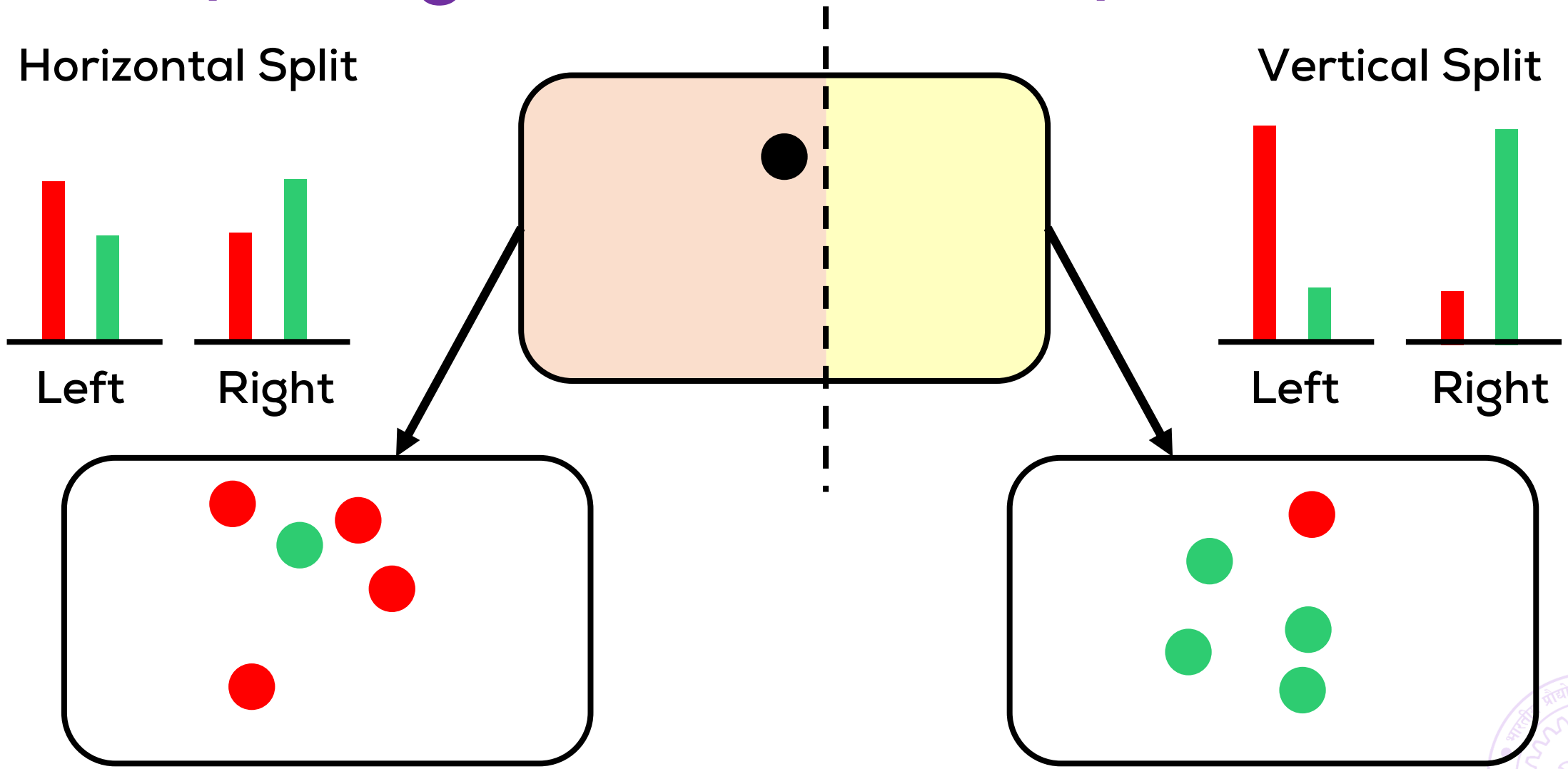
Node Splitting via feature stumps



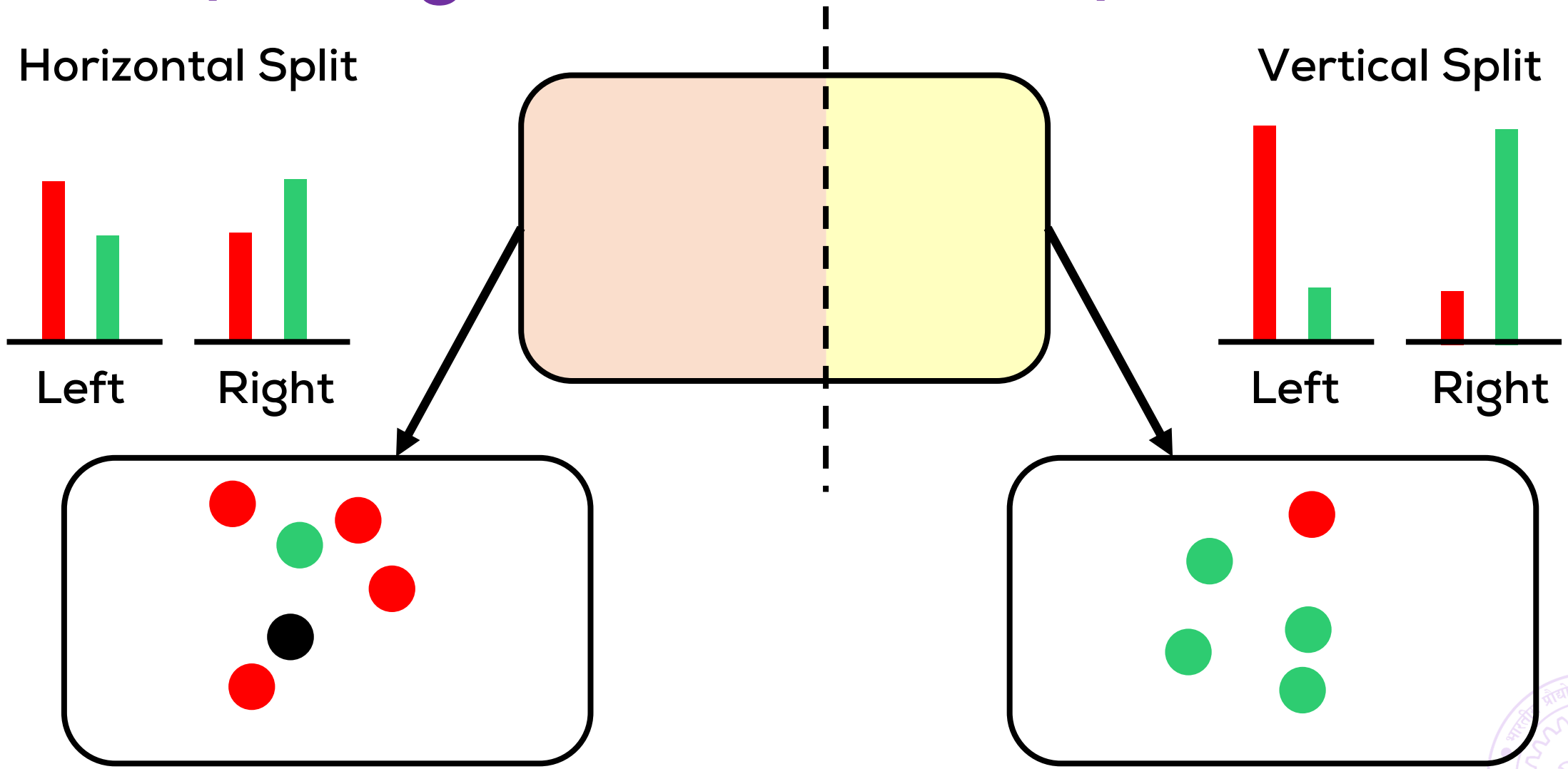
Node Splitting via feature stumps



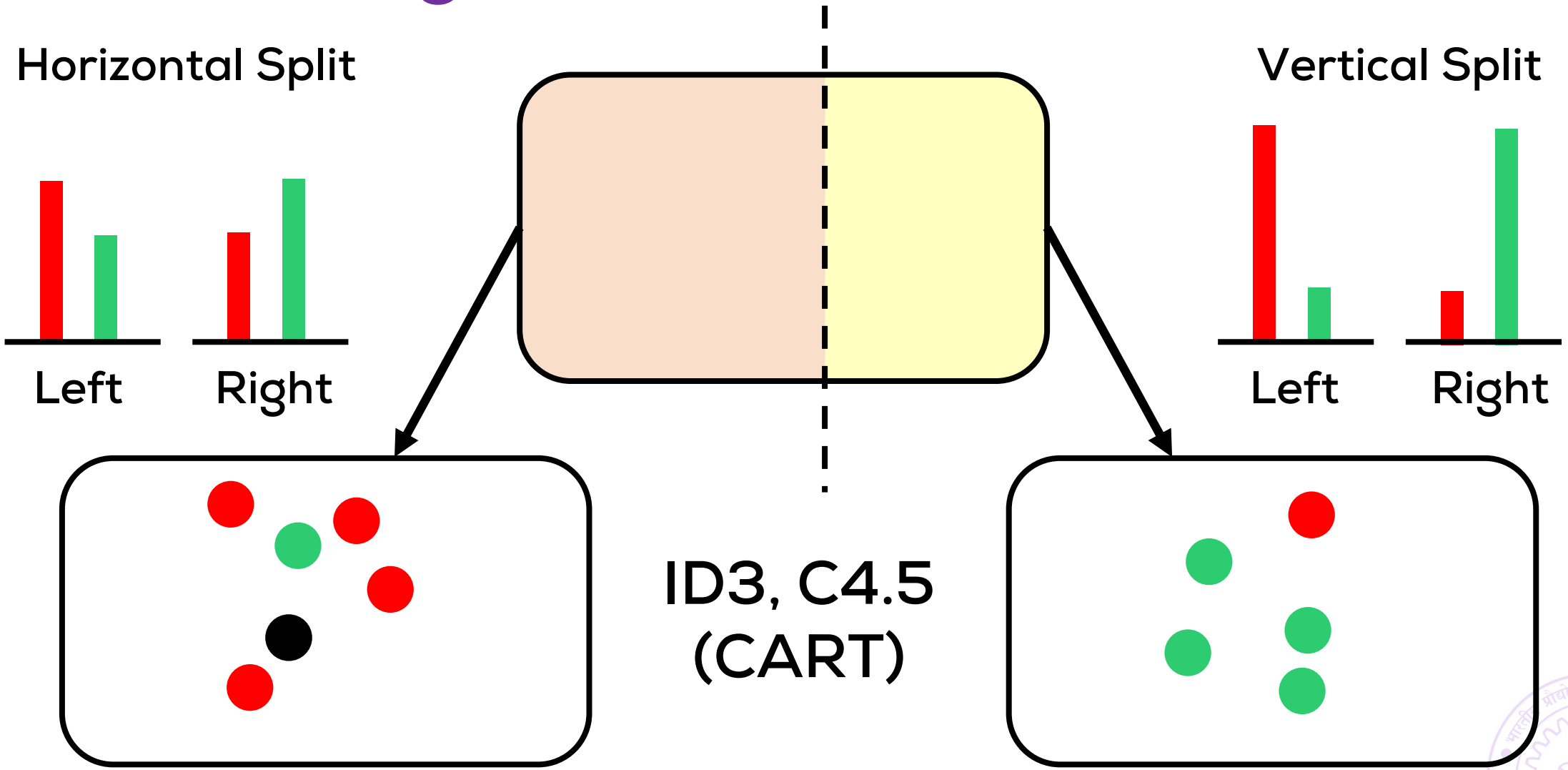
Node Splitting via feature stumps



Node Splitting via feature stumps

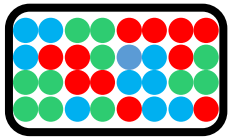


Node Splitting via feature stumps

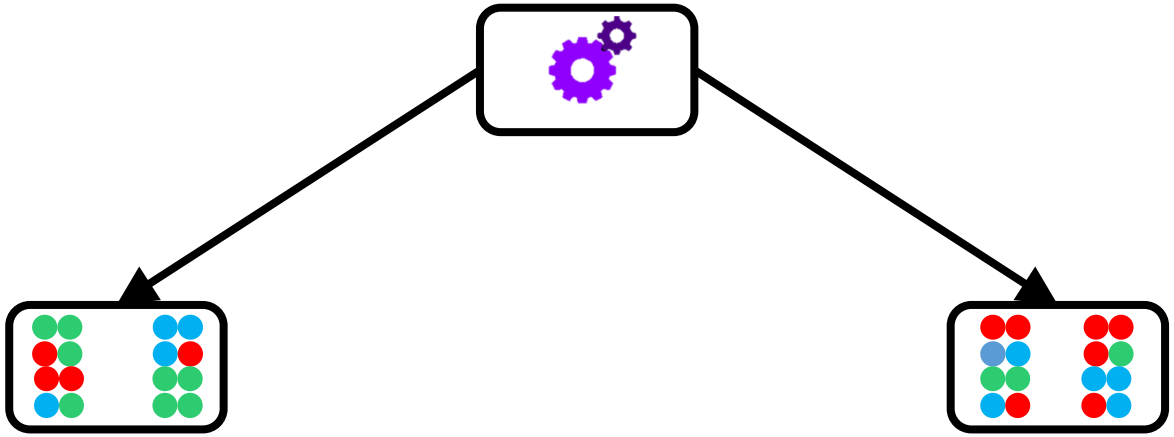


Exercise: Think about how to deal with regression!!

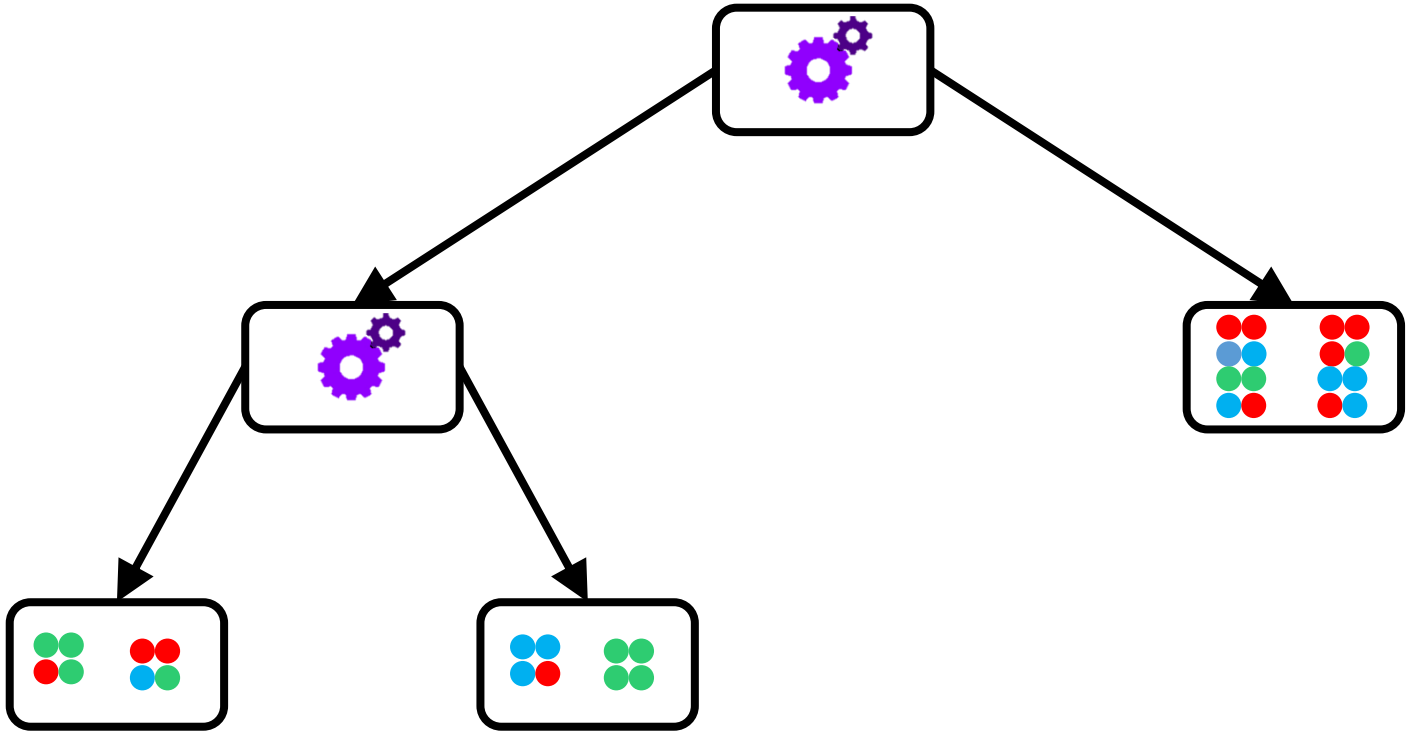
Growing the Tree



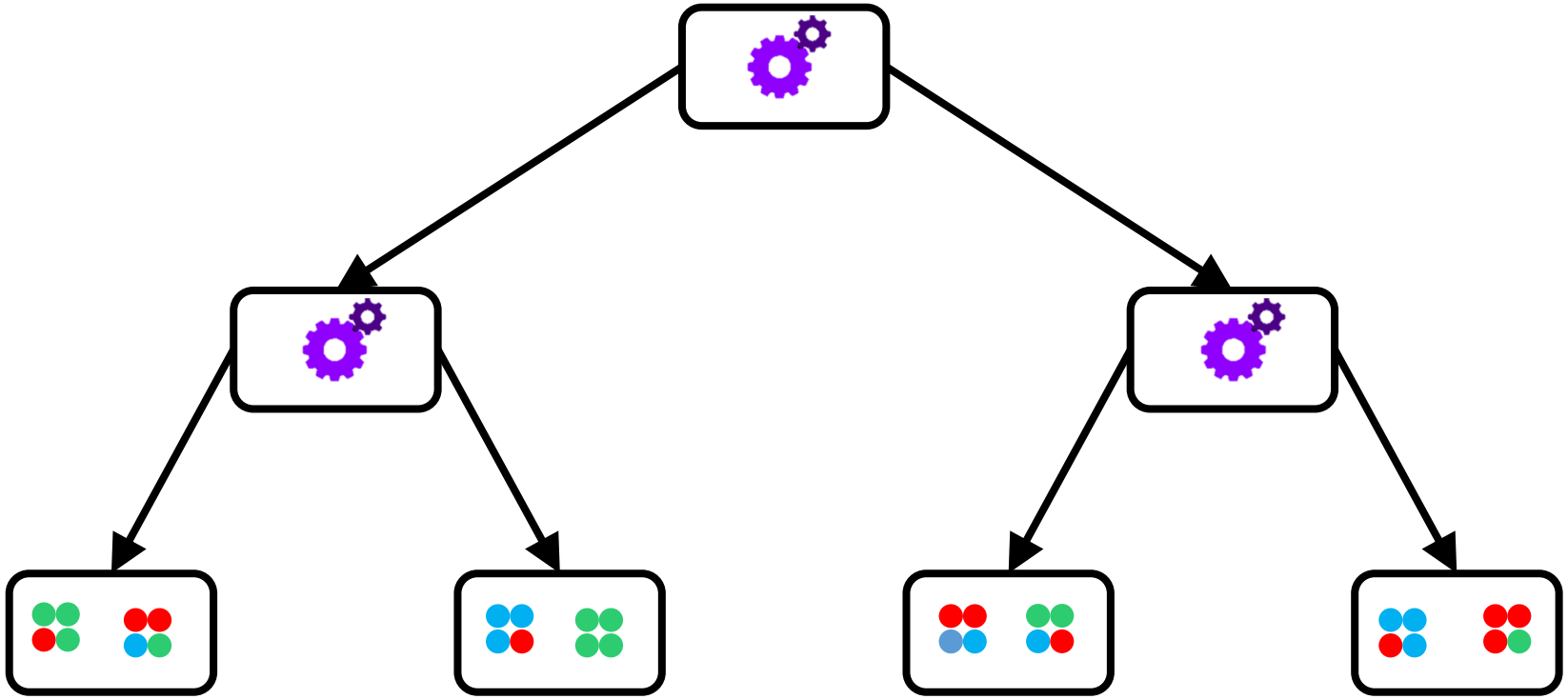
Growing the Tree



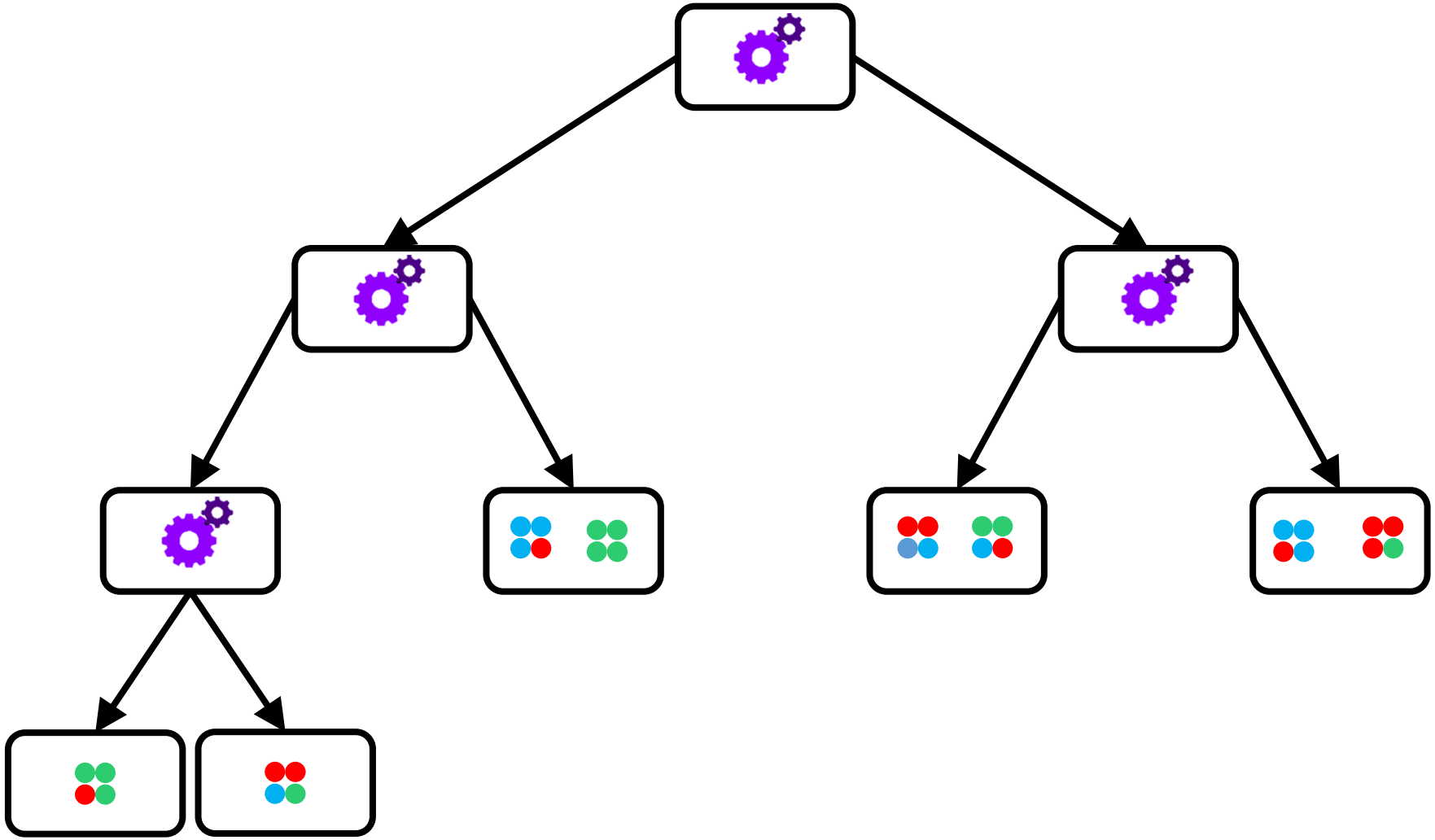
Growing the Tree



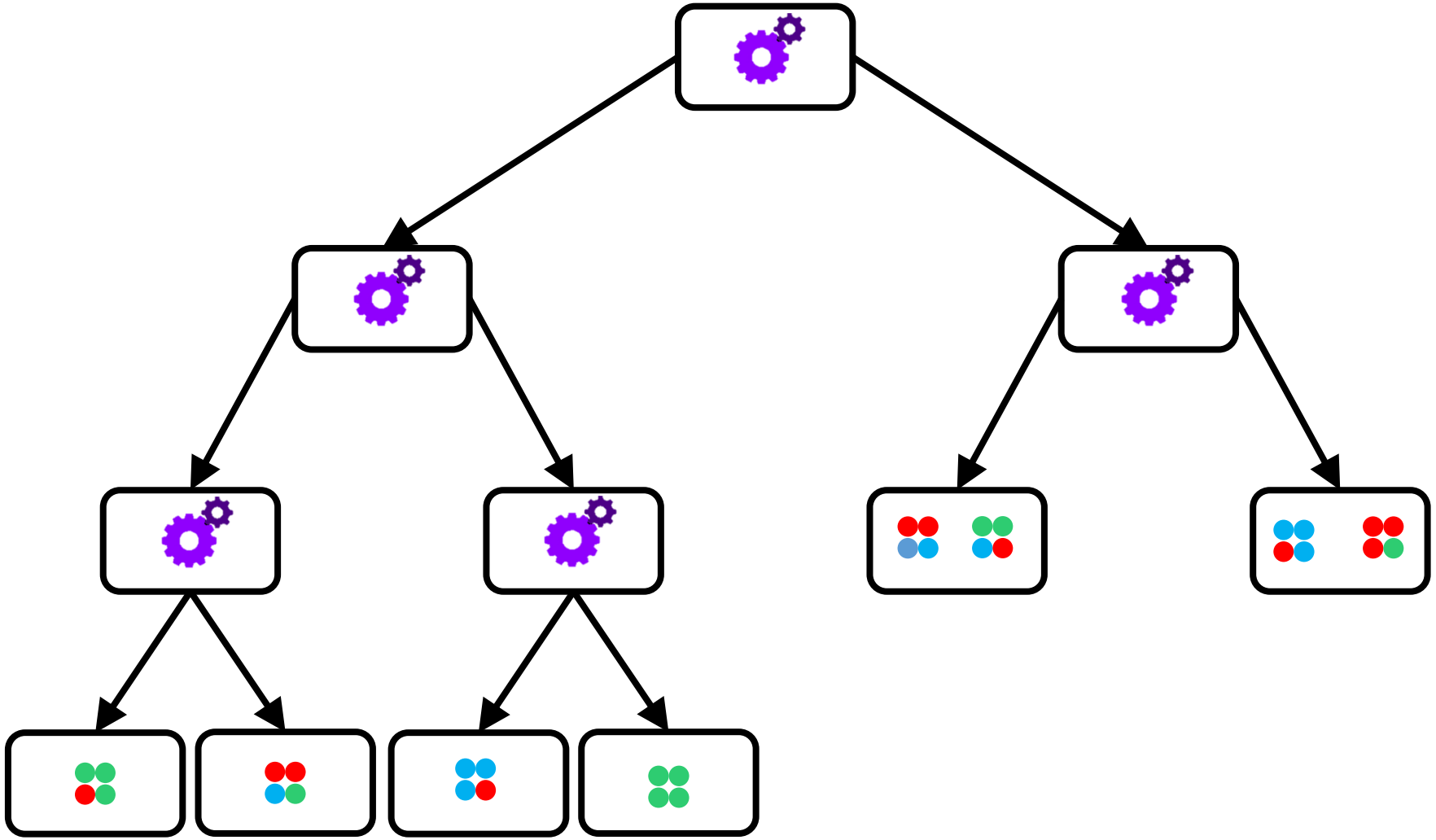
Growing the Tree



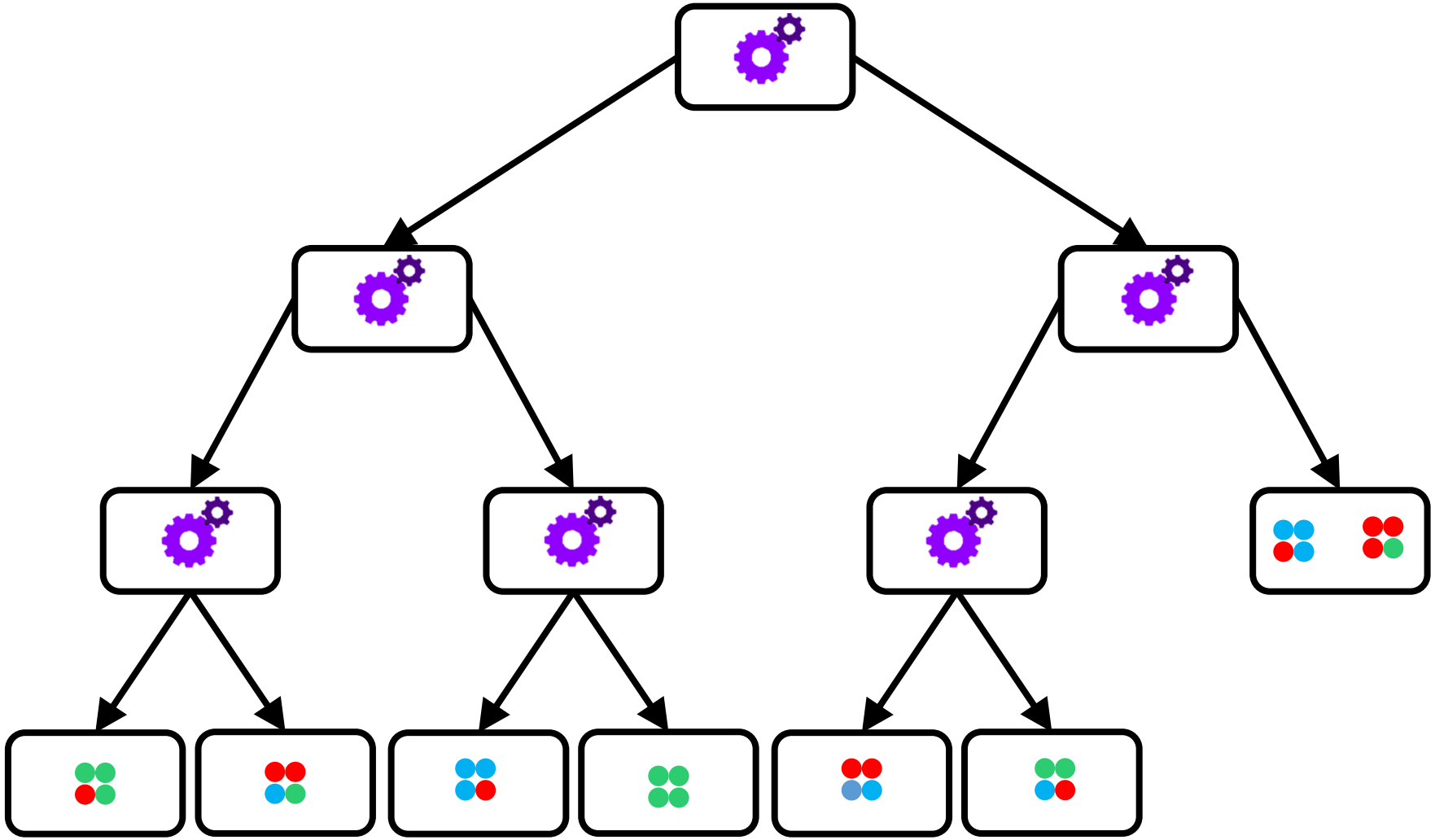
Growing the Tree



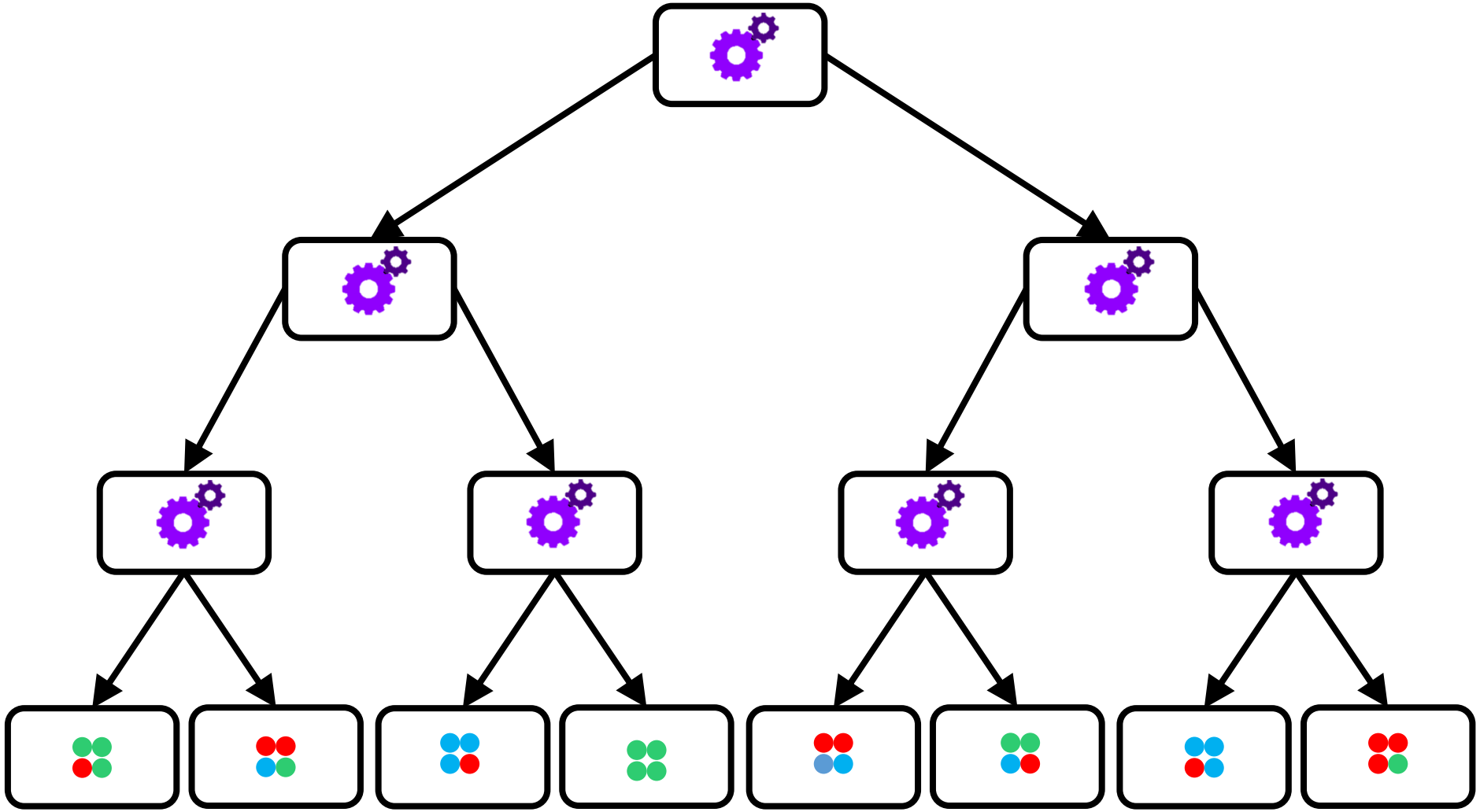
Growing the Tree



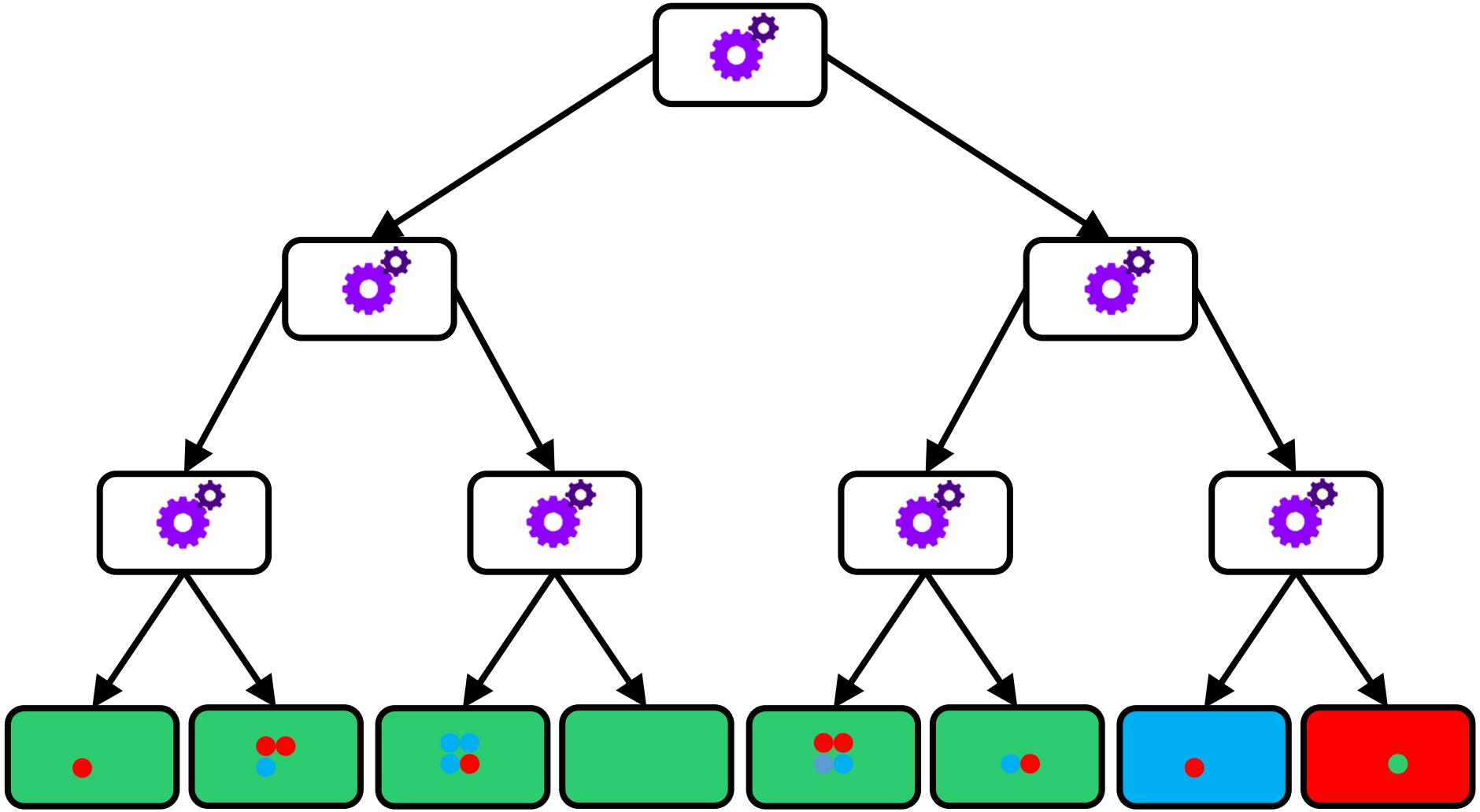
Growing the Tree



Growing the Tree

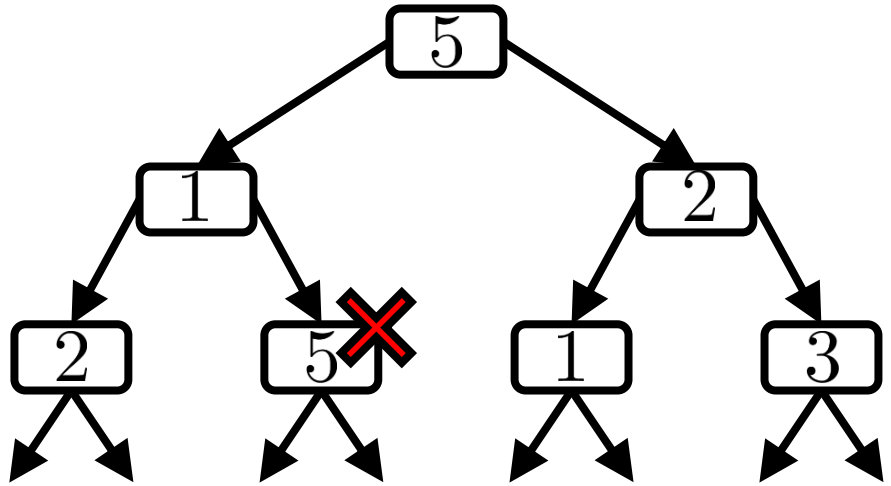


Growing the Tree



Pruning Strategies

- Stop if node is pure or almost pure
- Stop if all features exhausted
  - Don't use a feature twice on a path
  - Limits depth of tree to d
- Can stop if a node is ill-populated
- Can grow a tree and then merge nodes
  - Merge two leaves and see if it worsens or not - repeat
  - Use a validation set to make these decisions



A few Thoughts

- Why can't I use more features in a node decision stump?
  - Decision tree learning is an NP-hard problem in itself
  - Speed is an issue if using many features at each node
- Why cant I reuse features along a path?
  - Might mean you did not utilize it fully before
  - May lead to very deep trees
- Do I *have* to do things this way?
  - No, can experiment with well-founded, even crazy, ideas
  - But, give techniques that stood the test of time, a chance
- Some of the state-of-the-art multi-label classifiers are DTs

Please give your Feedback

http://tinyurl.com/ml17-18afb