CS 2050
Computer Science II

Thyago Mota

METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Agenda

- Binary Trees:
  - Decision Trees

Decision Trees

- A tree that supports decision models
- It maps observations of an item to conclusions about the item's target value
- It is one of the most widely used and practical methods for inductive inference

Decision Trees

```
1,30000,0,1,0
2,50000,0,1,2
4,70000,1,3,3
6,90000,1,3,3
2,55000,1,2,0
4,55000,1,2,3
3,60000,1,2,2
1,35000,0,1,2
1,25000,0,2,1
6,95000,1,3,4
6,85000,1,4,4
4,50000,1,3,3
3,50000,0,3,3
4,80000,1,3,2
6,90000,1,2,4
4,75000,1,3,4
2,60000,1,2,1
```

Decision Trees

of members

1	1,30000,0,1,0
2	2,50000,0,1,2
3	4,70000,1,3,3
4	6,90000,1,3,3
5	2,55000,1,2,0
6	4,55000,1,2,3
7	3,60000,1,2,2
8	1,35000,0,1,2
9	1,25000,0,2,1
10	6,95000,1,3,4
11	6,85000,1,4,4
12	4,50000,1,3,3
13	3,50000,0,3,3
14	4,80000,1,3,2
15	6,90000,1,2,4
16	4,75000,1,3,4
17	2,60000,1,2,1

Decision Trees

income

1	1,30000,0,1,0
2	2,50000,0,1,2
3	4,70000,1,3,3
4	6,90000,1,3,3
5	2,55000,1,2,0
6	4,55000,1,2,3
7	3,60000,1,2,2
8	1,35000,0,1,2
9	2,25000,0,2,1
10	6,95000,1,3,4
11	6,85000,1,4,4
12	4,50000,1,3,3
13	3,50000,0,3,3
14	4,80000,1,3,2
15	6,90000,1,2,4
16	4,75000,1,3,4
17	2,60000,1,2,1

Decision Trees

marital status
0: single, 1:married

1	1,30000,0,1,0
2	2,50000,0,1,2
3	4,70000,1,3,3
4	6,90000,1,3,3
5	2,55000,1,2,0
6	4,55000,1,2,3
7	3,60000,1,2,2
8	1,35000,0,1,2
9	1,25000,0,2,1
10	6,95000,1,3,4
11	6,85000,1,4,4
12	4,50000,1,3,3
13	3,50000,0,3,3
14	4,80000,1,3,2
15	6,90000,1,2,4
16	4,75000,1,3,4
17	2,60000,1,2,1

Decision Trees

FAQs read

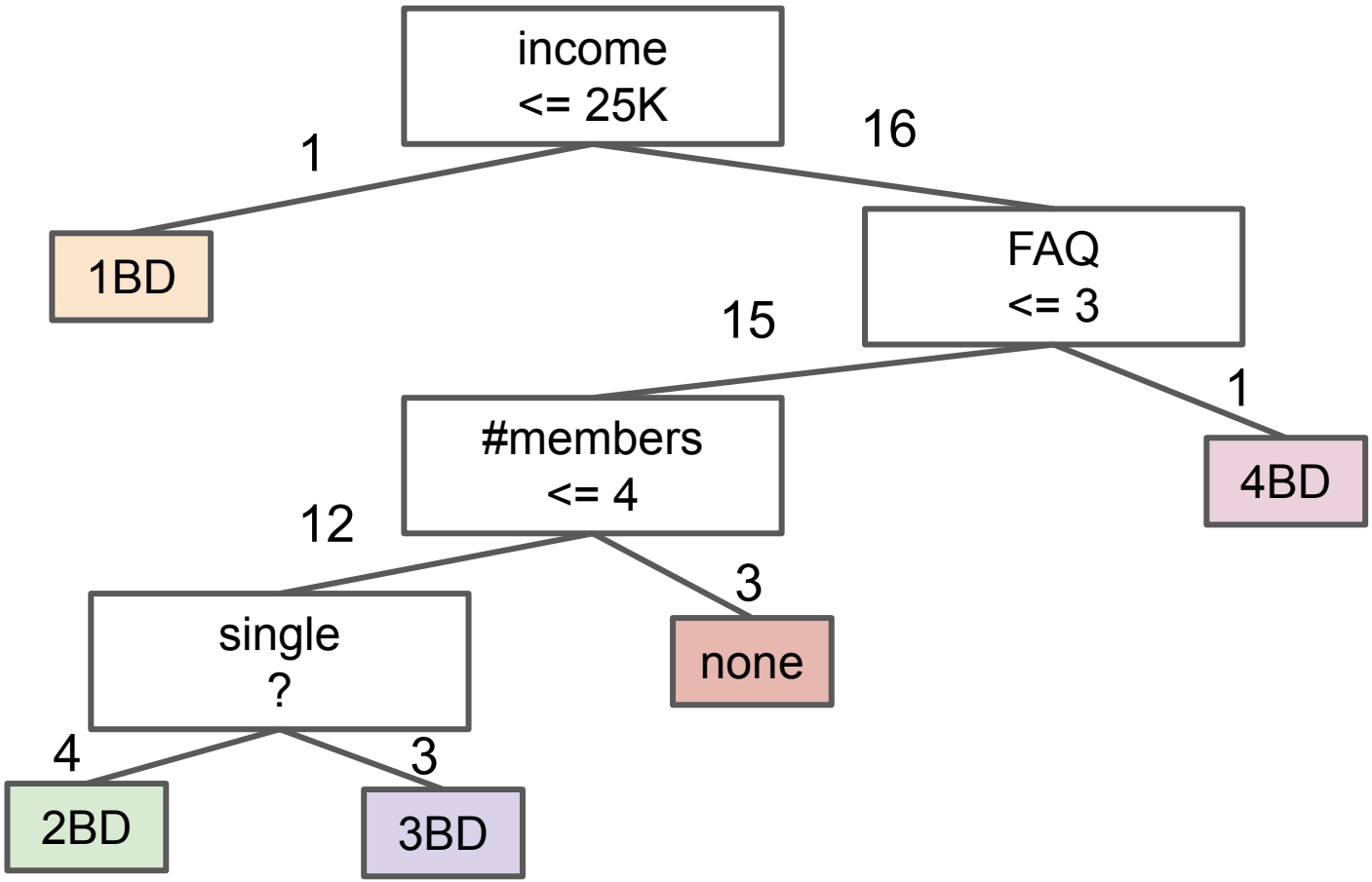
1	1,30000,0,1,0
2	2,50000,0,1,2
3	4,70000,1,3,3
4	6,90000,1,3,3
5	2,55000,1,2,0
6	4,55000,1,2,3
7	3,60000,1,2,2
8	1,35000,0,1,2
9	1,25000,0,2,1
10	6,95000,1,3,4
11	6,85000,1,4,4
12	4,50000,1,3,3
13	3,50000,0,3,3
14	4,80000,1,3,2
15	6,90000,1,2,4
16	4,75000,1,3,4
17	2,60000,1,2,1

Decision Trees

Decision
0: none, 1BD, 2BD, ...

1	1,30000,0,1,0
2	2,50000,0,1,2
3	4,70000,1,3,3
4	6,90000,1,3,3
5	2,55000,1,2,0
6	4,55000,1,2,3
7	3,60000,1,2,2
8	1,35000,0,1,2
9	1,25000,0,2,1
10	6,95000,1,3,4
11	6,85000,1,4,4
12	4,50000,1,3,3
13	3,50000,0,3,3
14	4,80000,1,3,2
15	6,90000,1,2,4
16	4,75000,1,3,4
17	2,60000,1,2,1

Decision Trees



Decision Trees

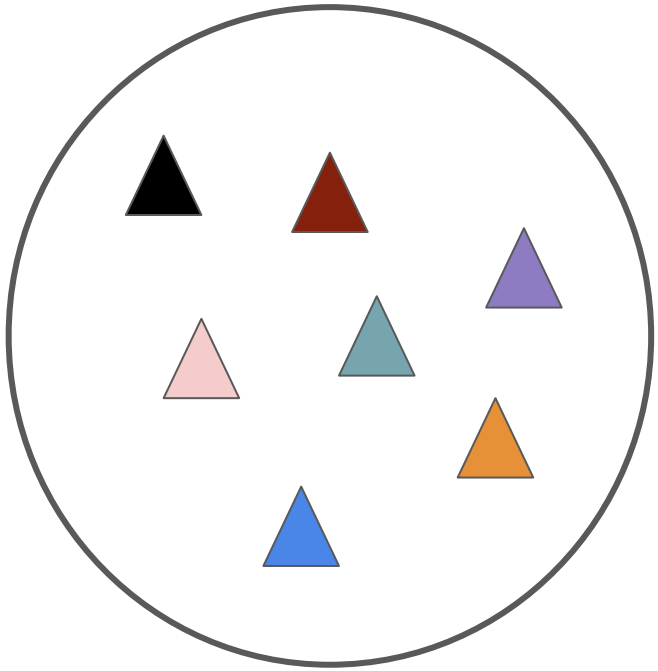
- Entropy:
  - A measurement of the randomness of a collection
  - The higher the entropy, the more disperse are the values in a collection
  - The lower the entropy, the more uniform are the values in a collection

Formula: $\sum_{i=1}^{c} - p_i \log_2 p_i$

Decision Trees

- Entropy:
  - Example 1:

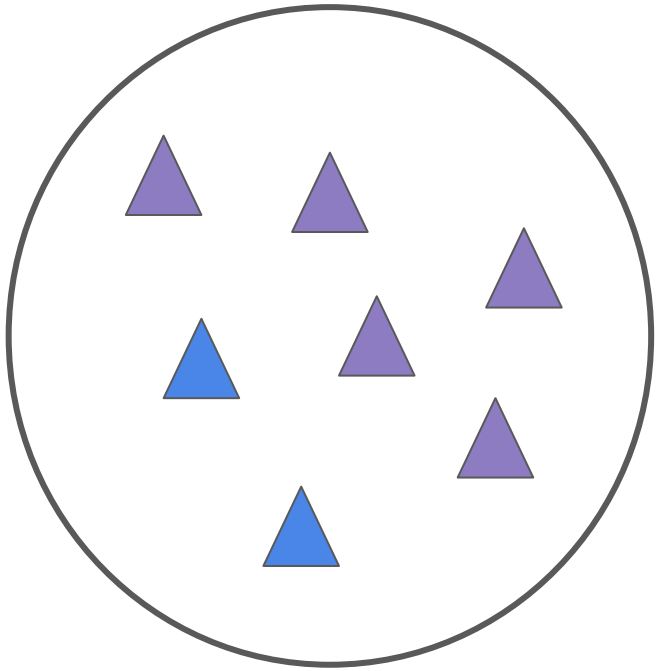
```
E = -1/7*log1/7 -1/7*log1/7
-1/7*log1/7 -1/7*log1/7
-1/7*log1/7 -1/7*log1/7
-1/7*log1/7 = 2.8079
```



Decision Trees

- Entropy:
  - Example 2:

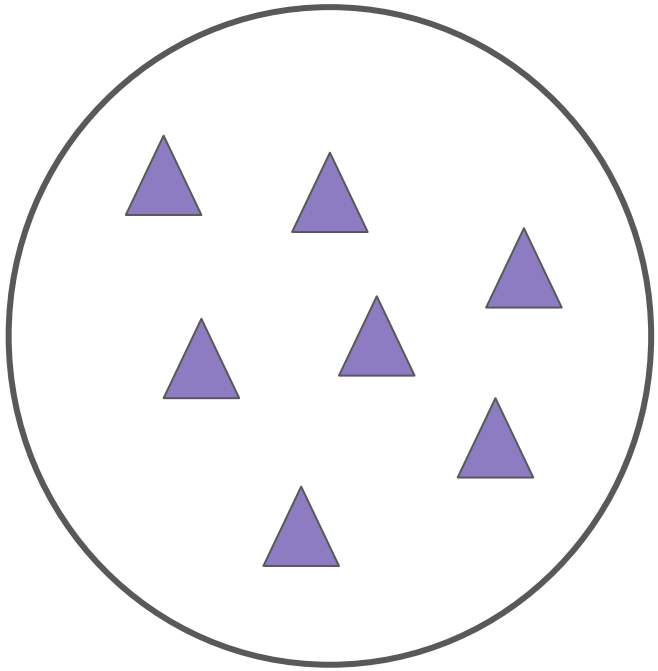
```
E = -5/7*log5/7 -2/7*log2/7

0.3467 + 0.5164 = 0.8631
```



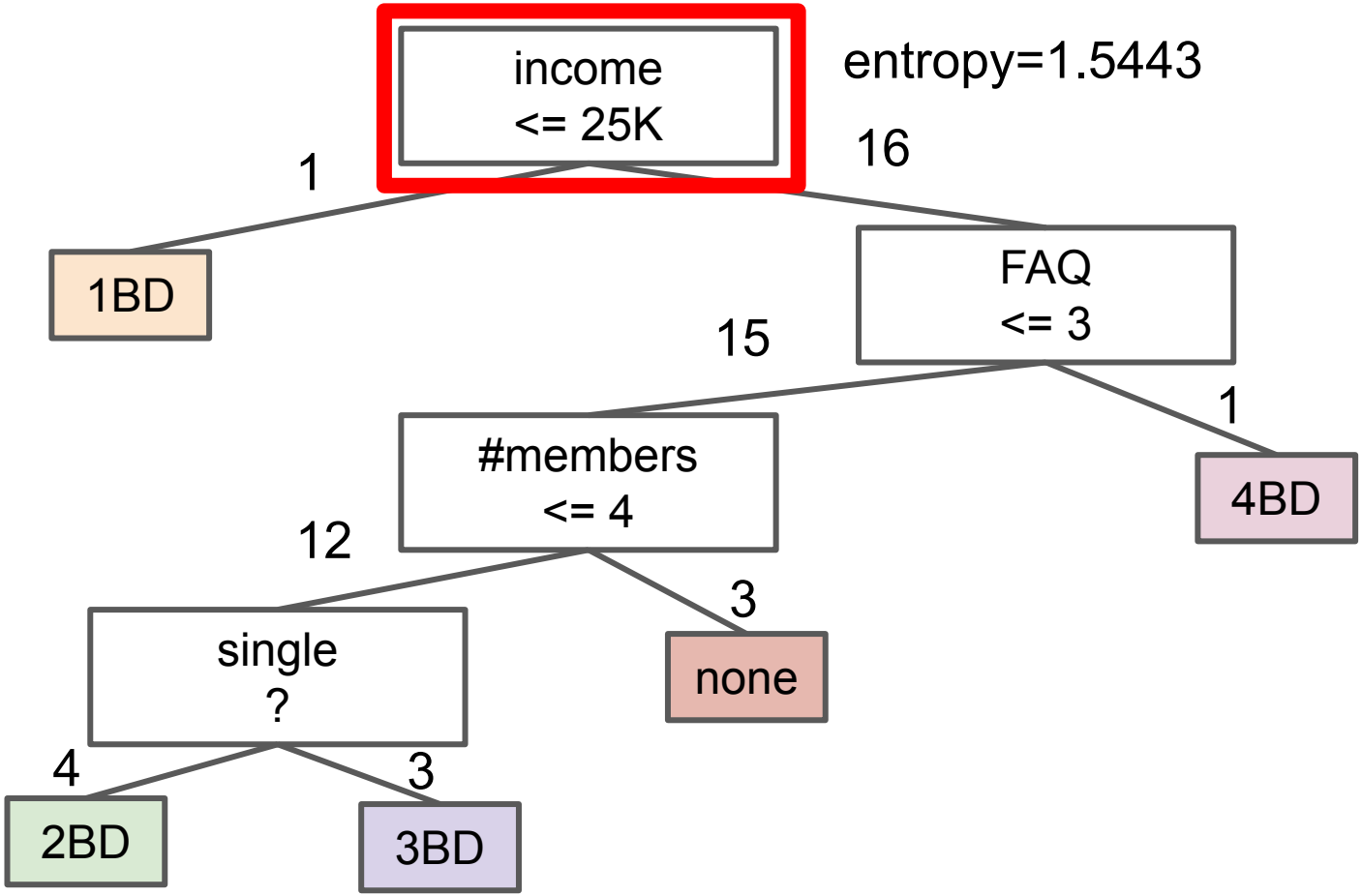
Decision Trees

- Entropy:
  - Example 3:

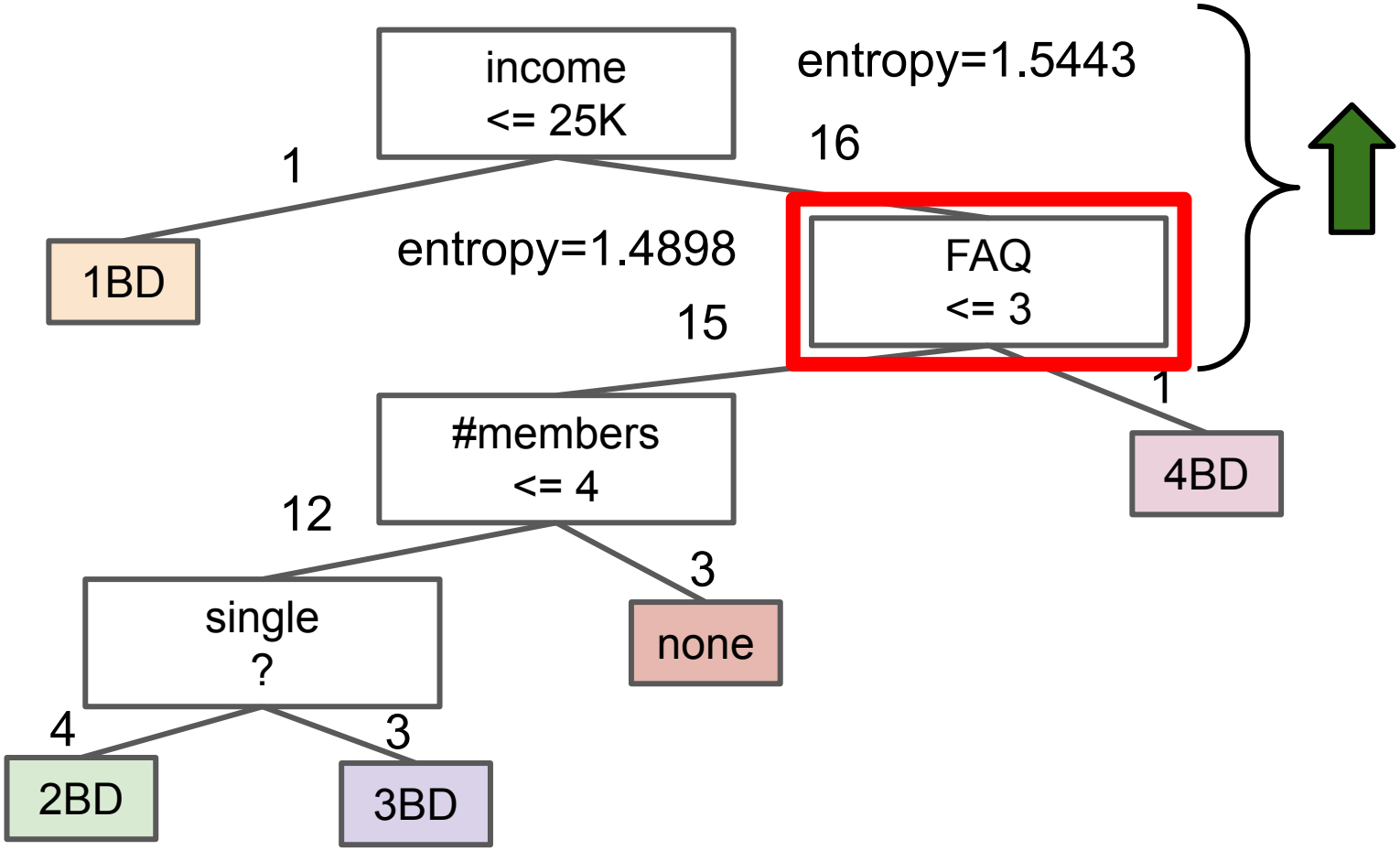
`E = -7/7*log7/7 = 0`



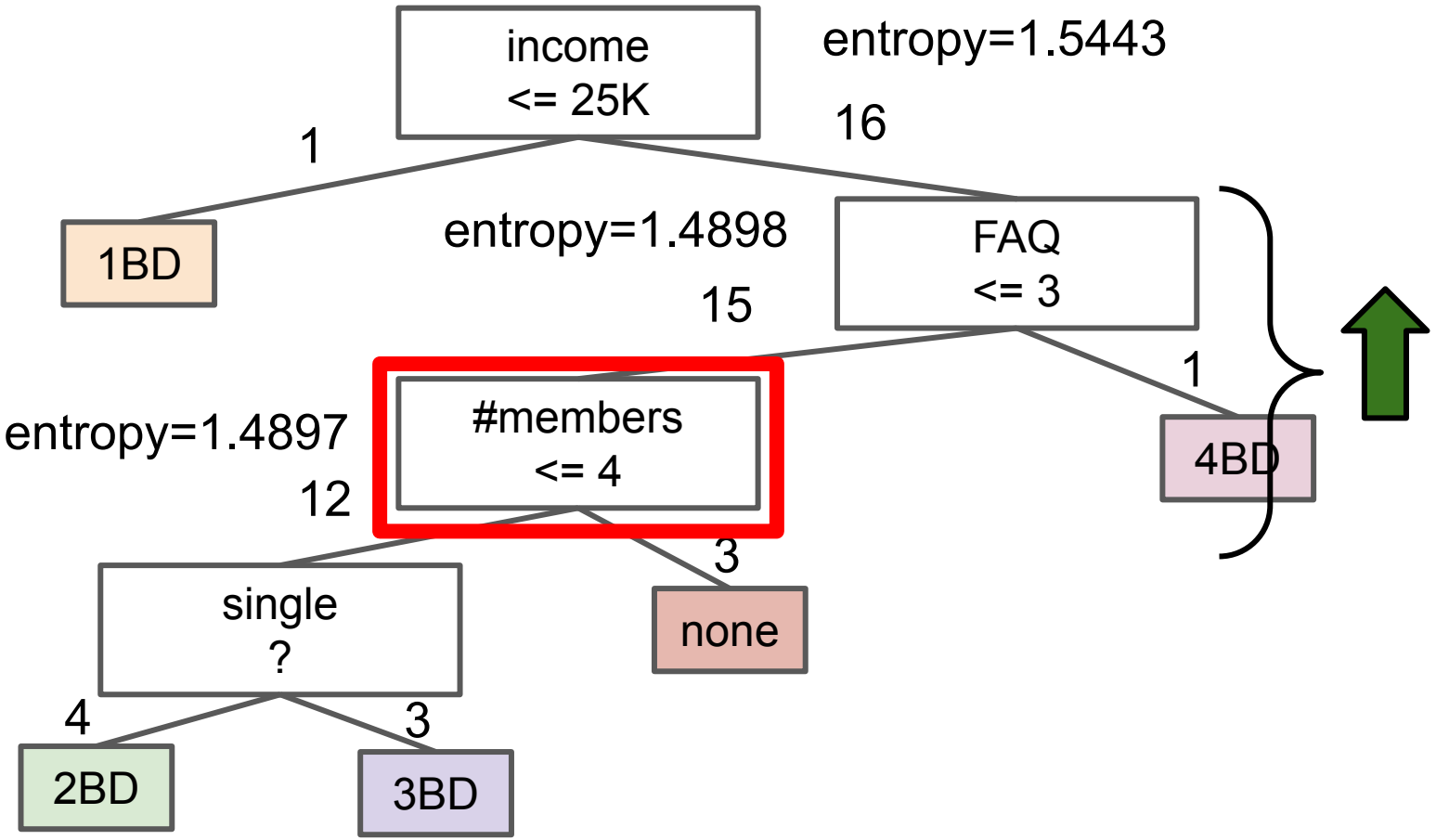
Decision Trees



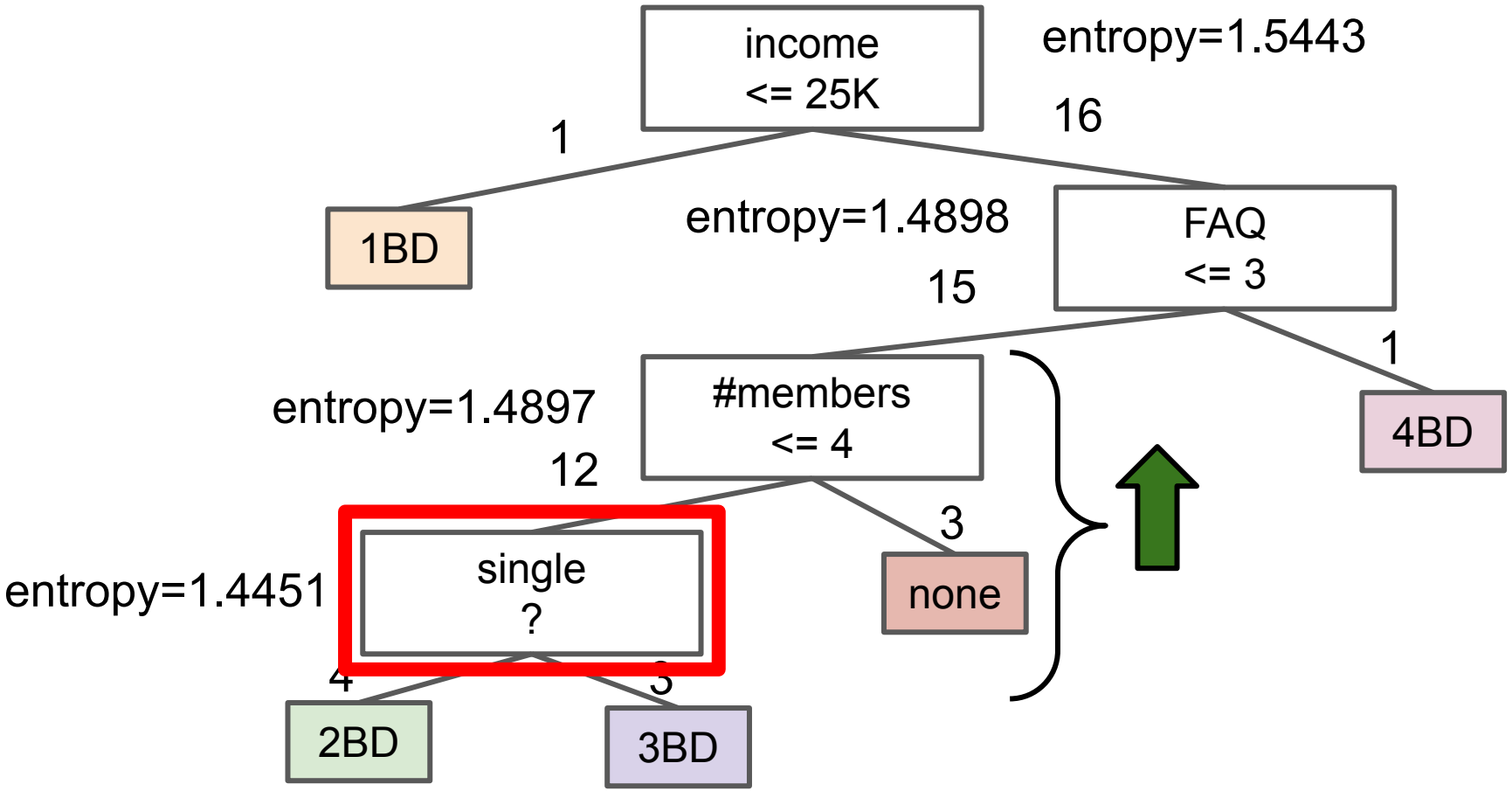
Decision Trees



Decision Trees



Decision Trees



Decision Trees

- ID3 (Iterative Dichotomiser 3) is an algorithm developed by Ross Quinlan to generate decision trees
- You can find the algorithm online at https://en.wikipedia.org/wiki/ID3_algorithm
- The code shared for this lesson has an implementation of ID3

Decision Trees

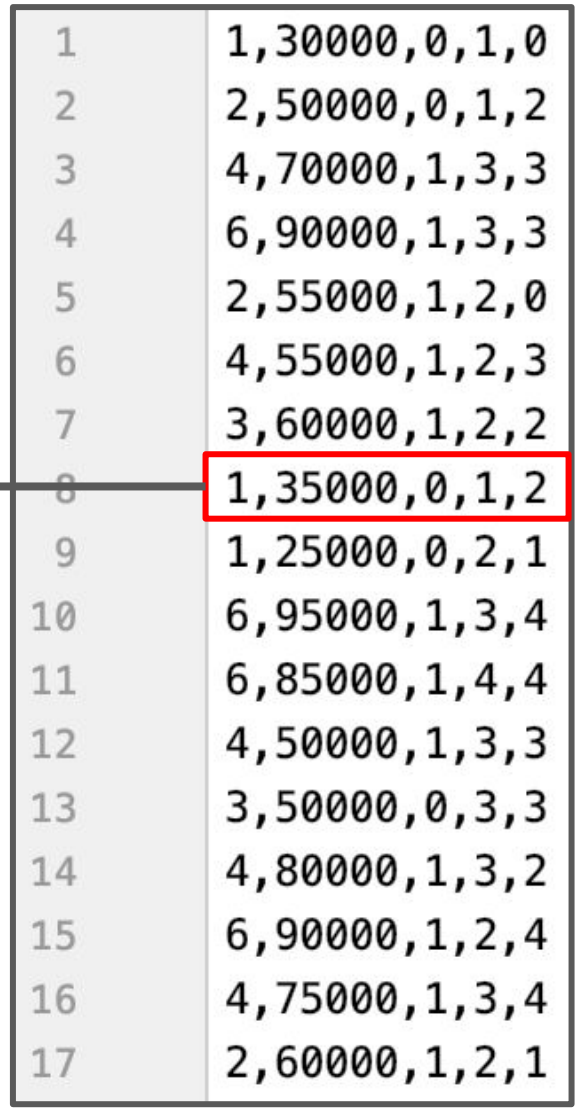
```
1,30000,0,1,0
2,50000,0,1,2
4,70000,1,3,3
6,90000,1,3,3
2,55000,1,2,0
4,55000,1,2,3
3,60000,1,2,2
1,35000,0,1,2
1,25000,0,2,1
6,95000,1,3,4
6,85000,1,4,4
4,50000,1,3,3
3,50000,0,3,3
4,80000,1,3,2
6,90000,1,2,4
4,75000,1,3,4
2,60000,1,2,1
```

Decision Trees

target

[1, 35000, 0, 1] 2

attributes



1	1,30000,0,1,0
2	2,50000,0,1,2
3	4,70000,1,3,3
4	6,90000,1,3,3
5	2,55000,1,2,0
6	4,55000,1,2,3
7	3,60000,1,2,2
8	1,35000,0,1,2
9	1,25000,0,2,1
10	6,95000,1,3,4
11	6,85000,1,4,4
12	4,50000,1,3,3
13	3,50000,0,3,3
14	4,80000,1,3,2
15	6,90000,1,2,4
16	4,75000,1,3,4
17	2,60000,1,2,1

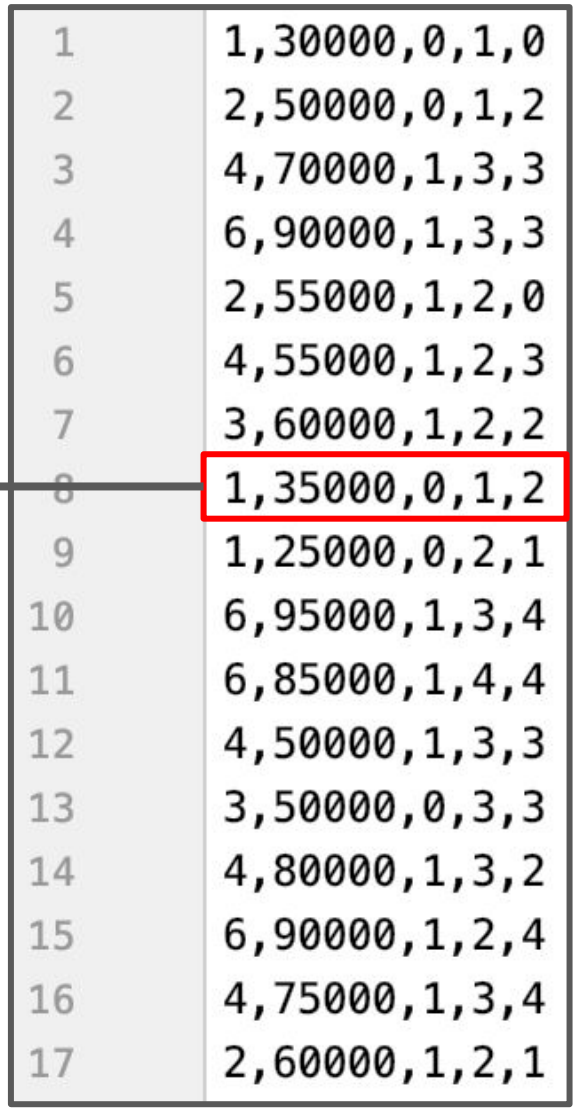
Decision Trees

target

[1, 35000, 0, 1] 2

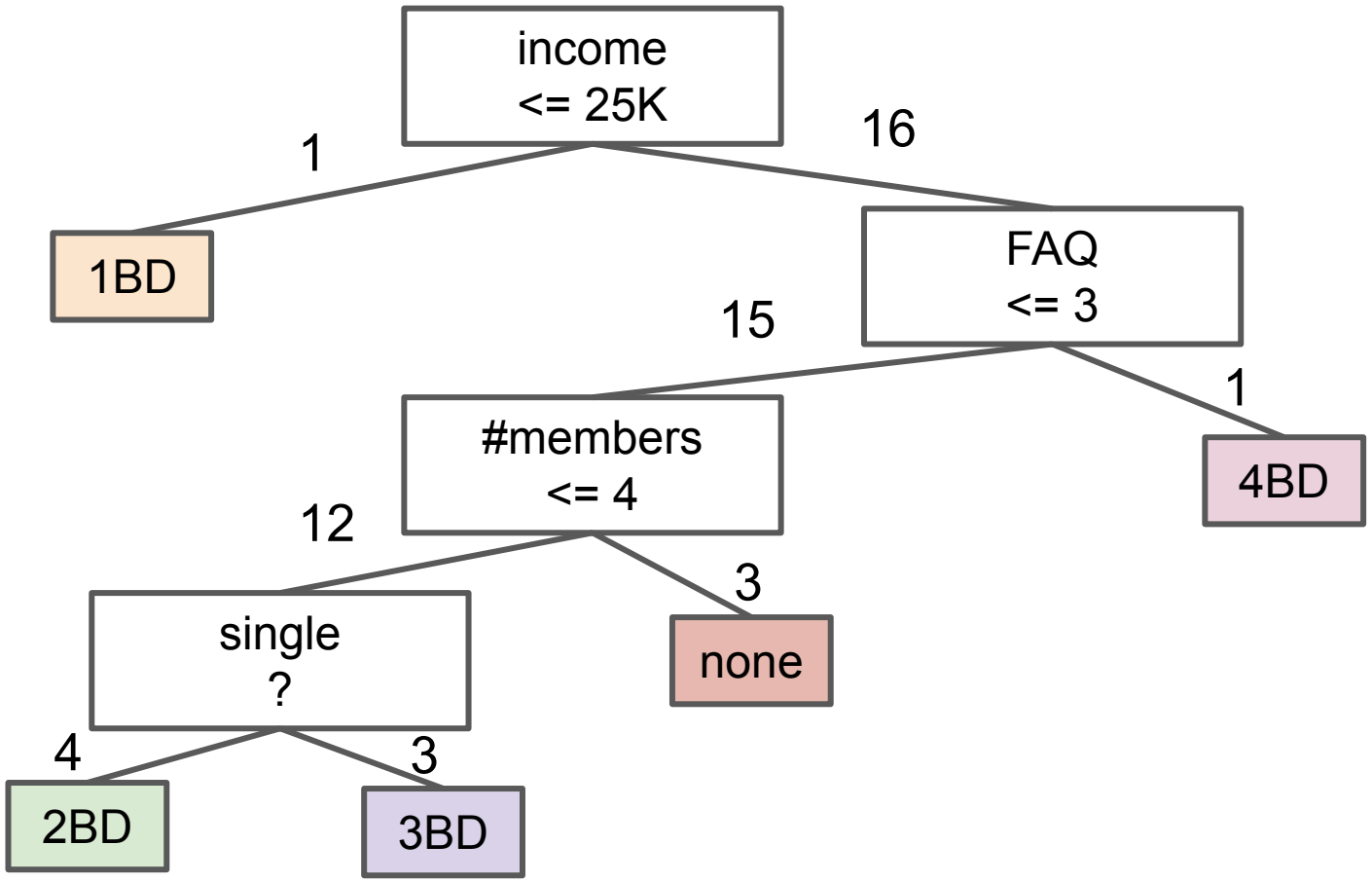
attributes

DataPoint
double attributes[];
int target;
int numberTargets;



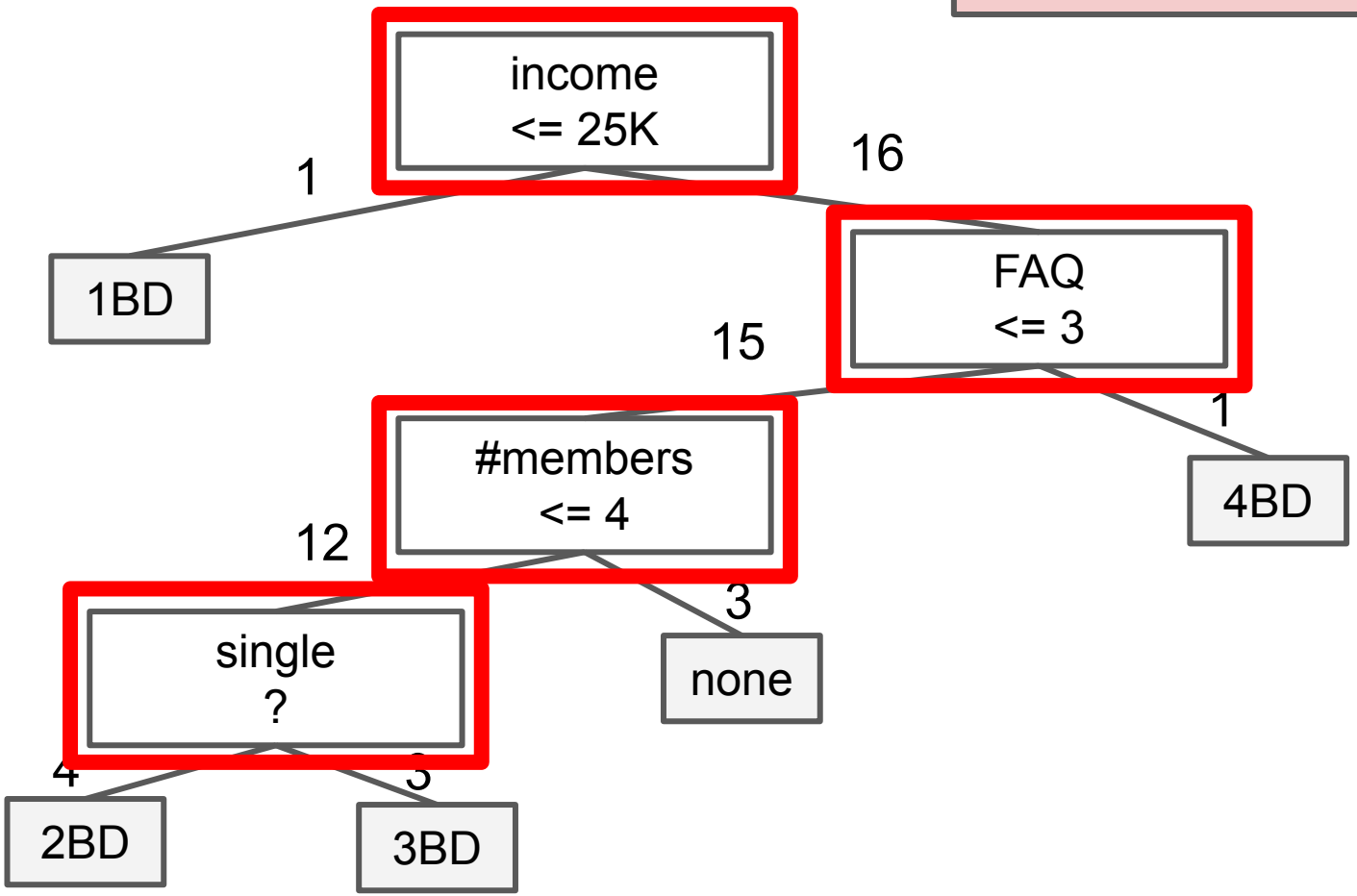
```
1,30000,0,1,0
2,50000,0,1,2
4,70000,1,3,3
6,90000,1,3,3
2,55000,1,2,0
4,55000,1,2,3
3,60000,1,2,2
1,35000,0,1,2
1,25000,0,2,1
6,95000,1,3,4
6,85000,1,4,4
4,50000,1,3,3
3,50000,0,3,3
4,80000,1,3,2
6,90000,1,2,4
4,75000,1,3,4
2,60000,1,2,1
```

Decision Trees



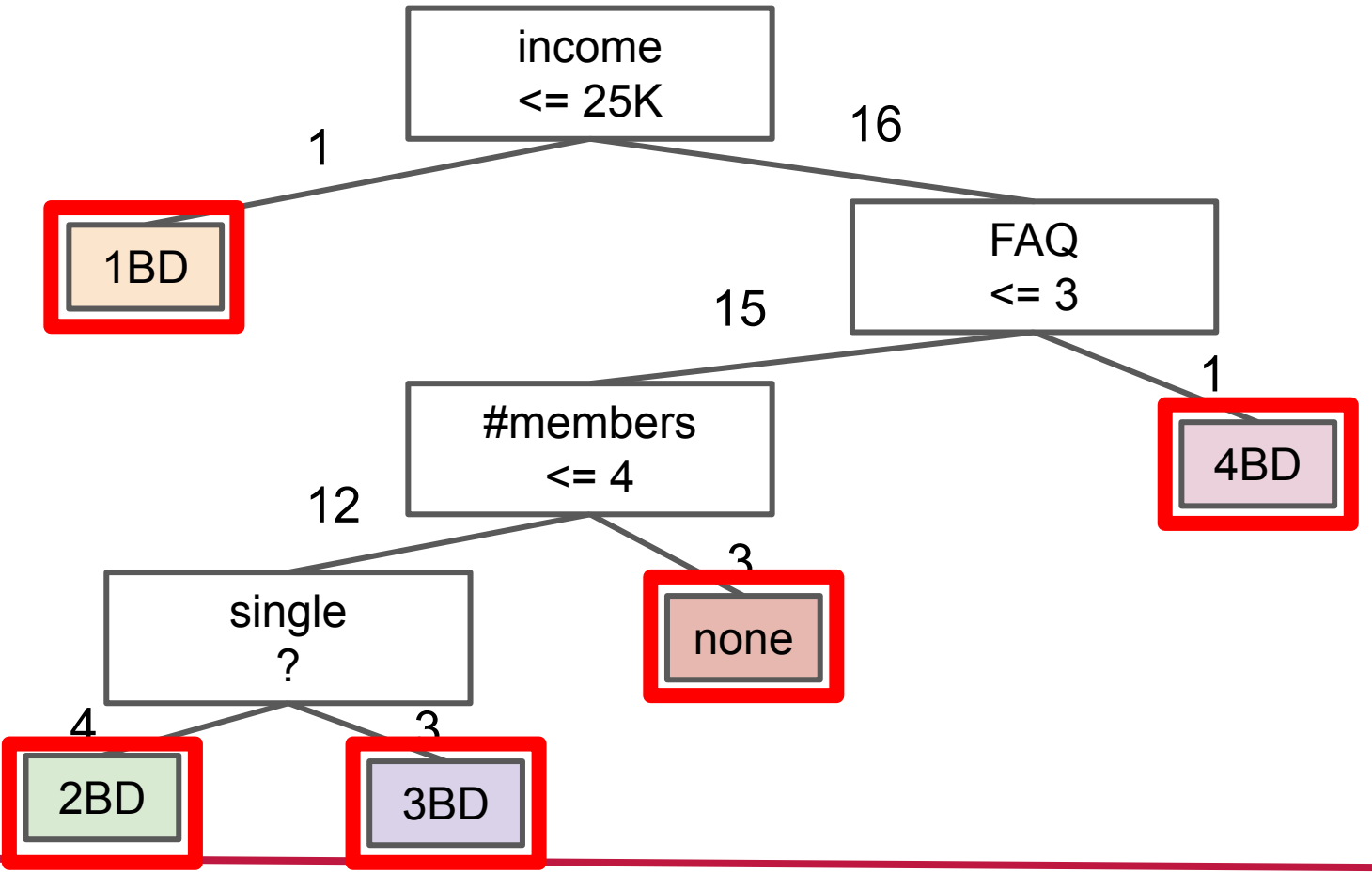
Decision Trees

Decision Data

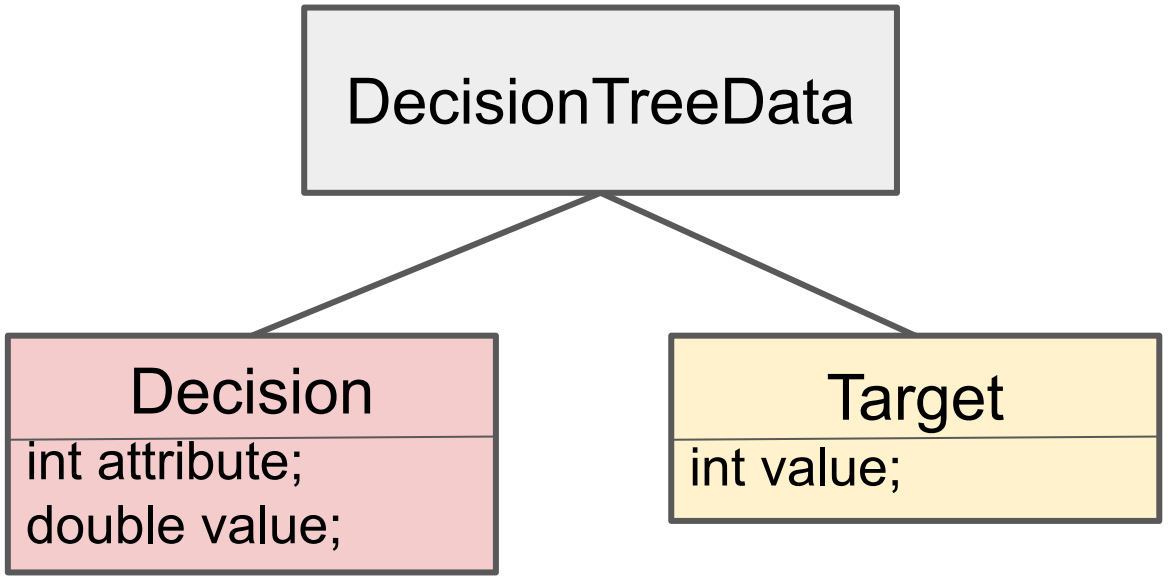


Decision Trees

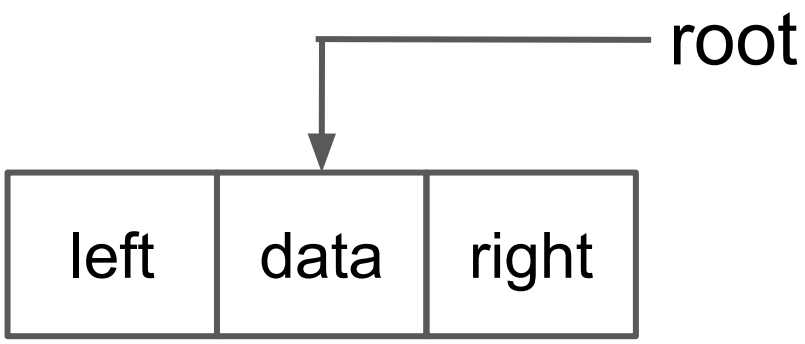
Target Data



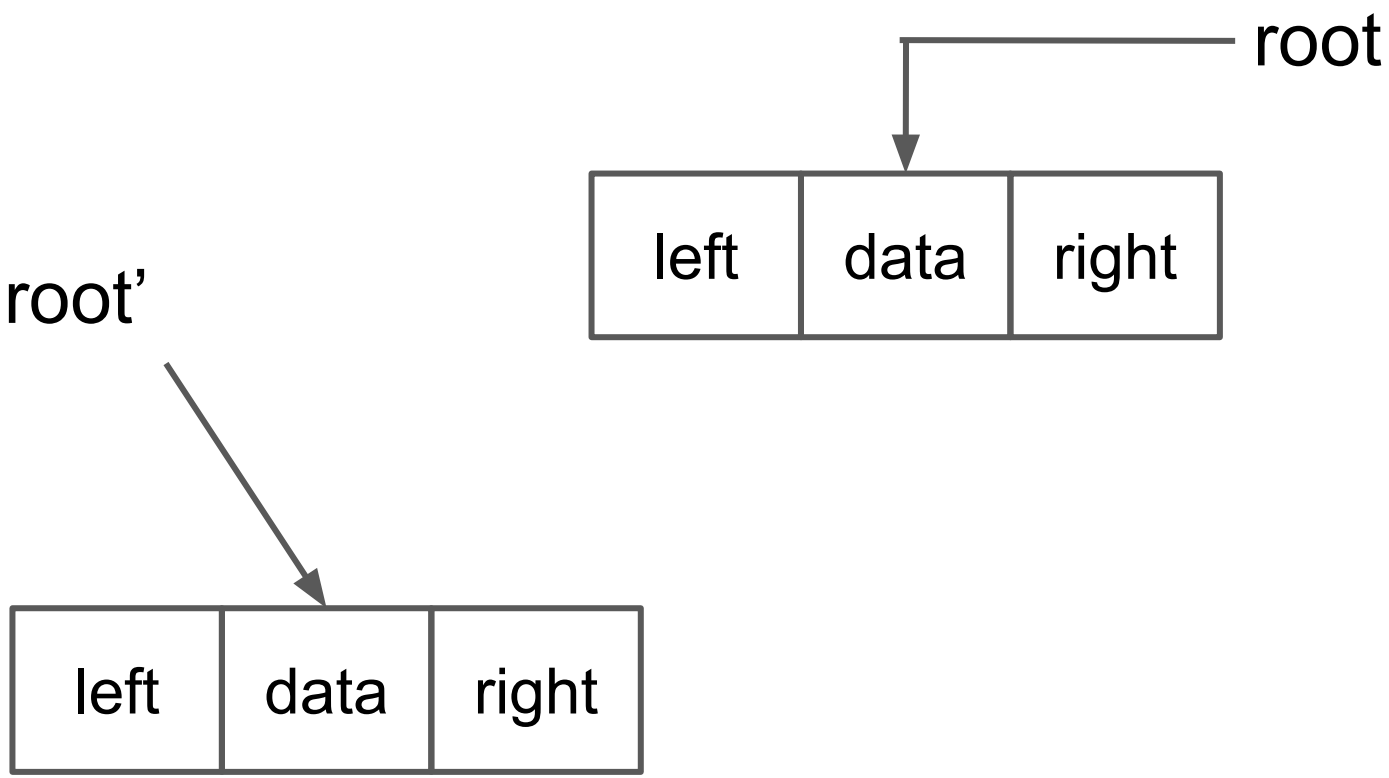
Decision Trees



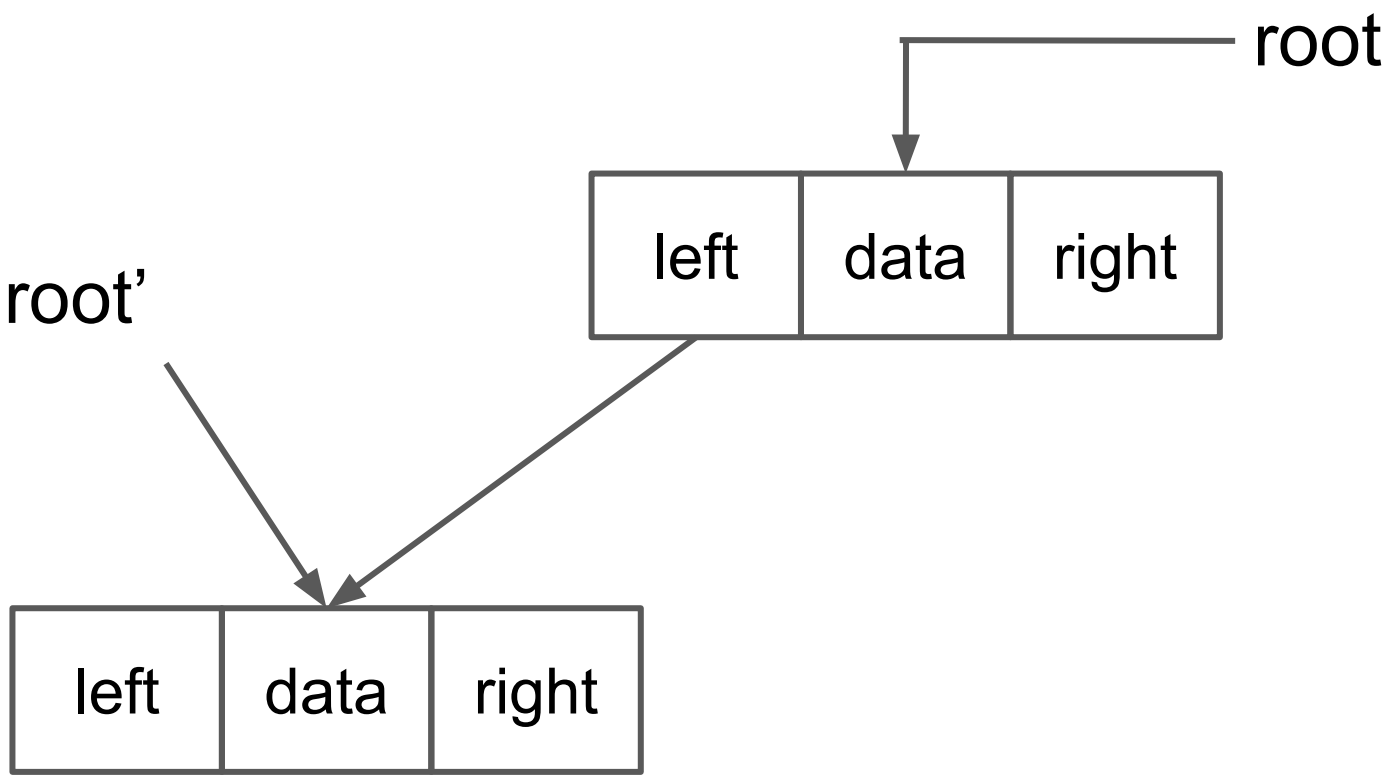
Decision Trees



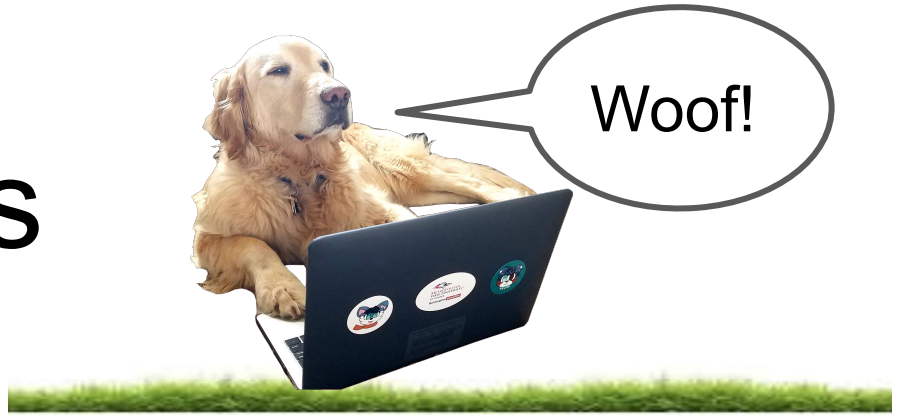
Decision Trees



Decision Trees



Decision Trees



- Practice #1:
  - Implement the **`classifyRecursively`** method in the **`DecisionTree`** class
  - This method takes a reference to a **`DecisionTreeNode`** and a **`DataPoint`** object to classify
  - Internal nodes carry **`Decision`** objects while or leaf nodes carry **`Target`** objects

```java
// TODO: finish implementation of classifyRecursively
private void classifyRecursively(DecisionTreeNode current, DataPoint dp) {

    // TODO: get DecisionTreeData from current DecisionTreeNode
    DecisionTreeData data = current.getData();

    // TODO: if DecisionTreeData is an instance of Decision, cast it to Decision and
    // use the object to decide whether to call classifyRecursively to the left or to the right
    if (data instanceof Decision) {
        Decision decision = (Decision) data;
        int attribute = decision.getAttribute();
        double value = decision.getValue();
        if (dp.getAttribute(attribute) <= value)
            classifyRecursively(current.getLeft(), dp);
        else
            classifyRecursively(current.getRight(), dp);
    }

    // TODO: if DecisionTreeData is an instance of Target, cast it to Target and
    // set the target of the data point
    else {
        Target target = (Target) data;
        dp.setTarget(target.getValue());
    }
}
```

Decision Trees

- Practice #2:
  - Try to create other decision trees
  - The Kaggle community shares many datasets that you can use to start building your own decision trees
  - Link: http://kaggle.com