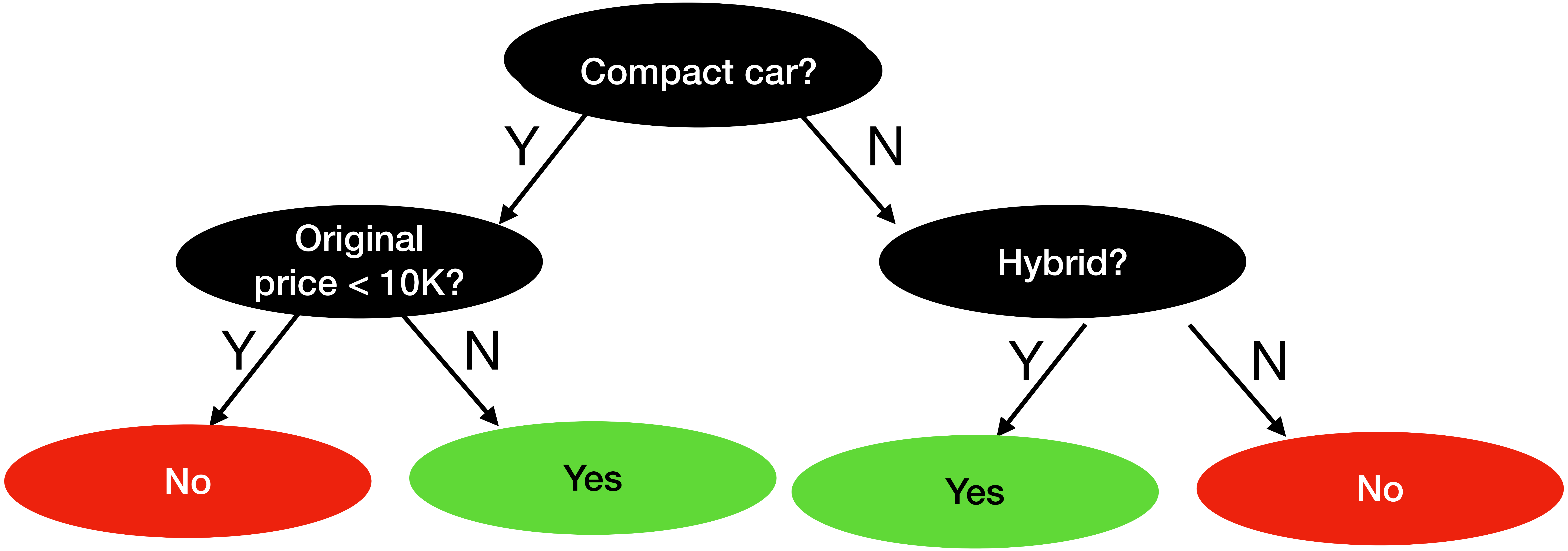
Decision Tree, finished product

To classify/predict: will the car be driven more than 200000 miles?



Training data: examples with feature values, correct class

Compact	Hybrid	Price	Class
No	No	9K	-
No	No	8K	-
No	No	7K	-
Yes	No	6K	-
No	No	10K	+
Yes	No	15K	+
No	Yes	17K	+
No	Yes	20K	+

+: Driven over 200,000 miles

-: Not driven over 200,000 miles

Transforming categorical & numerical features to Boolean "yes or no" questions

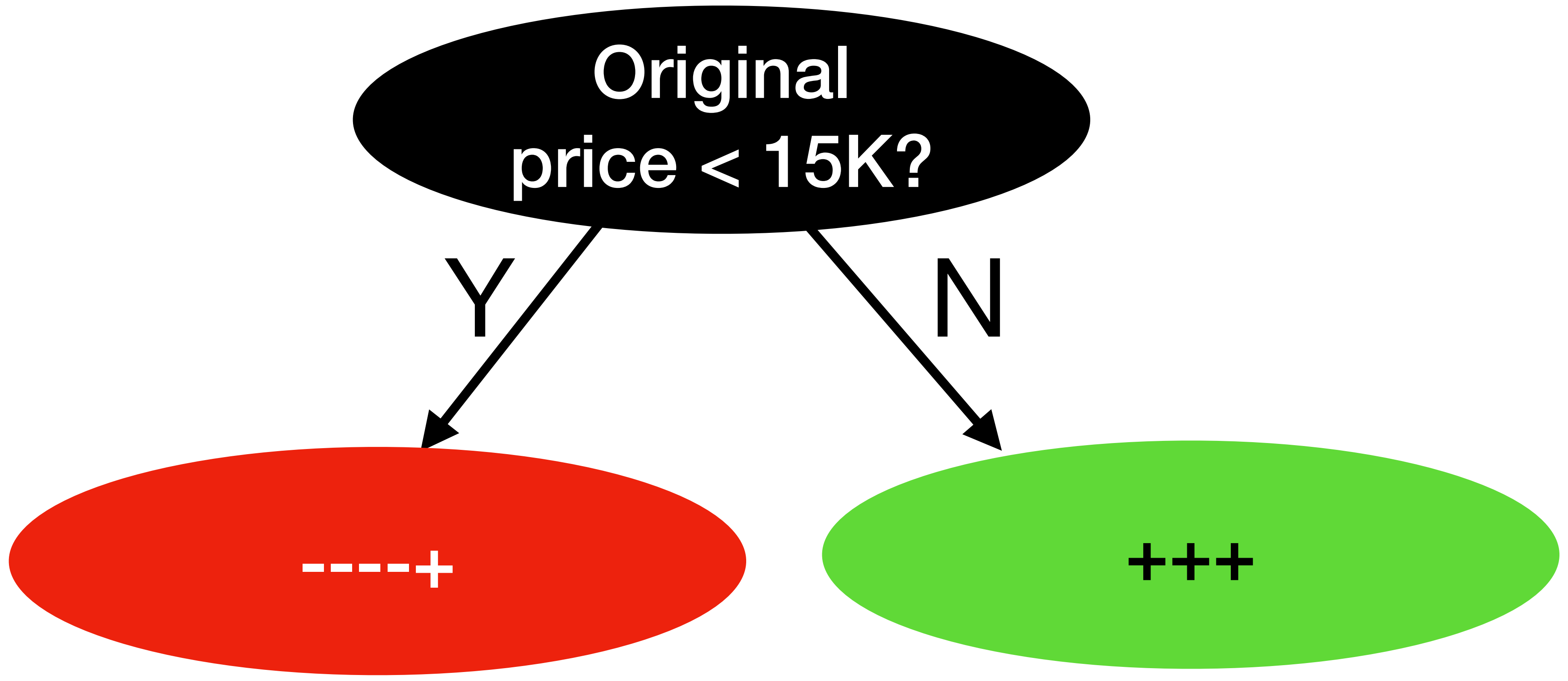
- **Categorical features (e.g., make of car):** Becomes one question/feature per value of the categorical feature: "Is it a Corolla? Is it a Fit?"
  - This strategy is generally known as "one-hot encoding."
- **Numerical features (e.g., price of car):** Becomes one Boolean "threshold" question per observed value: "Did it cost < 10K? Did it cost < 15K?"

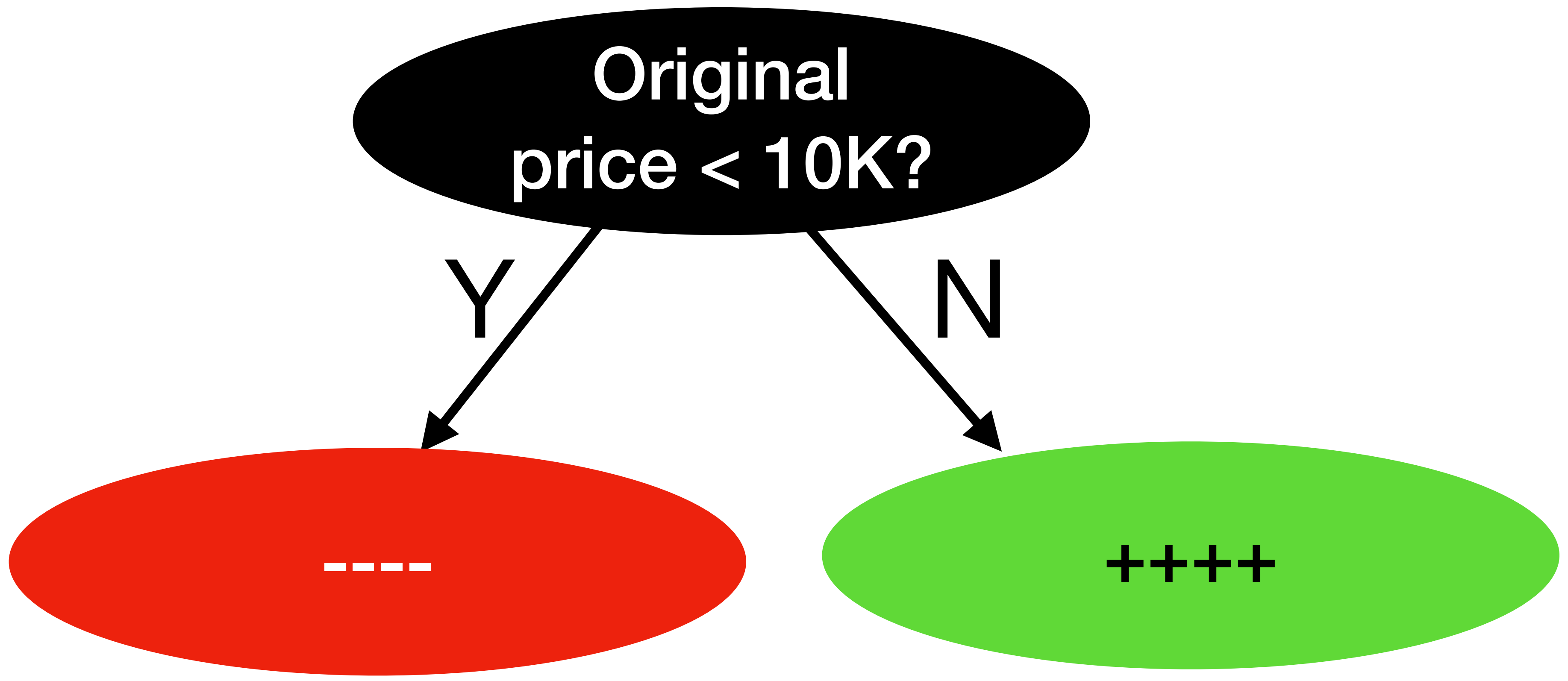
Strategy: find questions that make the labels agree within "pile"

Price	Class
9K	-
8K	-
7K	-
6K	-
20K	+
17K	+
15K	+
10K	+

+: Driven over 200,000 miles

-: Not driven over 200,000 miles

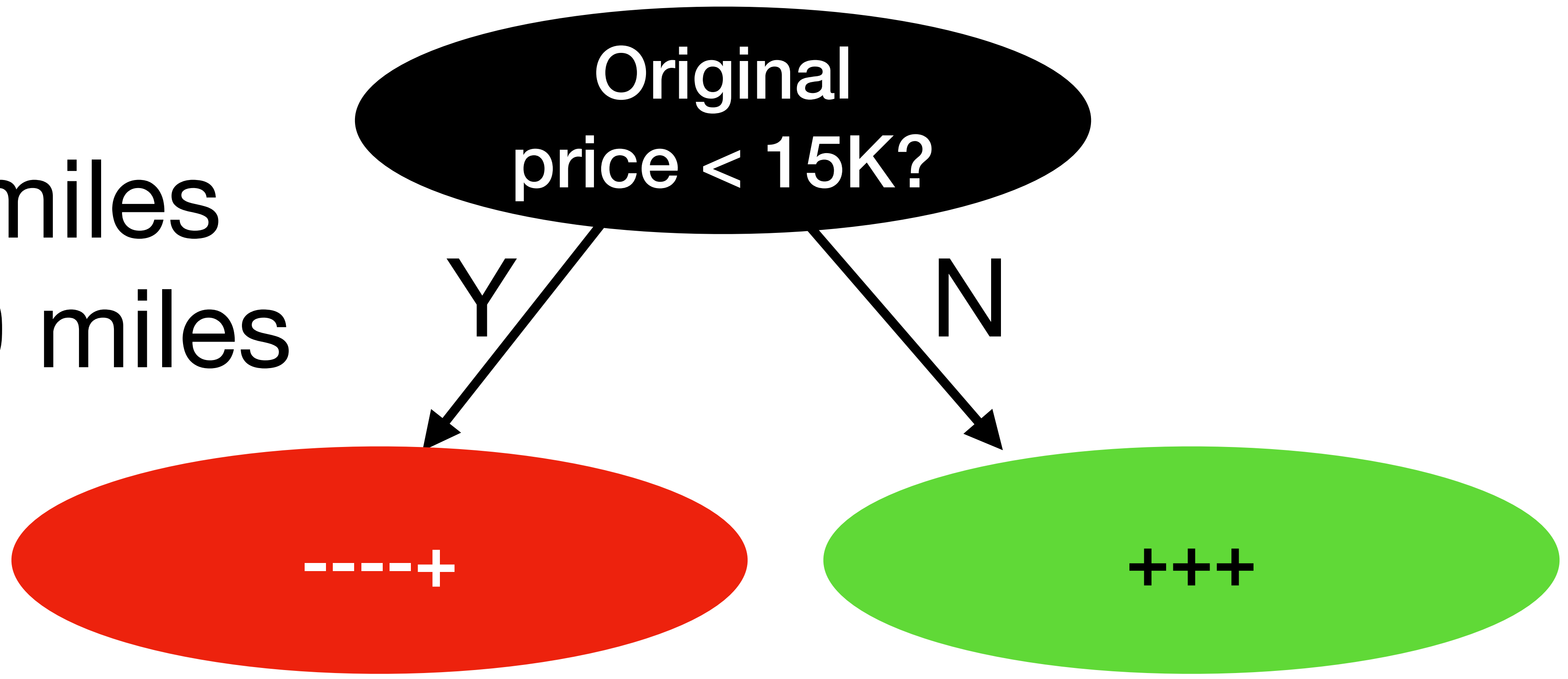




Reminder: the - and +'s are labeled data points

+: Driven over 200,000 miles

-: Not driven over 200,000 miles



Compact	Hybrid	Price	Class
No	No	9K	-
No	No	8K	-
No	No	7K	-
Yes	No	6K	-
Yes	No	10K	+

Compact	Hybrid	Price	Class
Yes	No	15K	+
No	Yes	17K	+
No	Yes	20K	+

Entropy: $\sum_i -p_i \log_2 p_i$ where p_i is the probability of symbol

++++

Entropy 0

0 lg 0 - 1 lg 1 = 0 - 0 = 0

++--

Entropy 1

-1/2 lg 1/2 - 1/2 lg 1/2 = - 1/2(-1) - 1/2(-1) = 1

+---

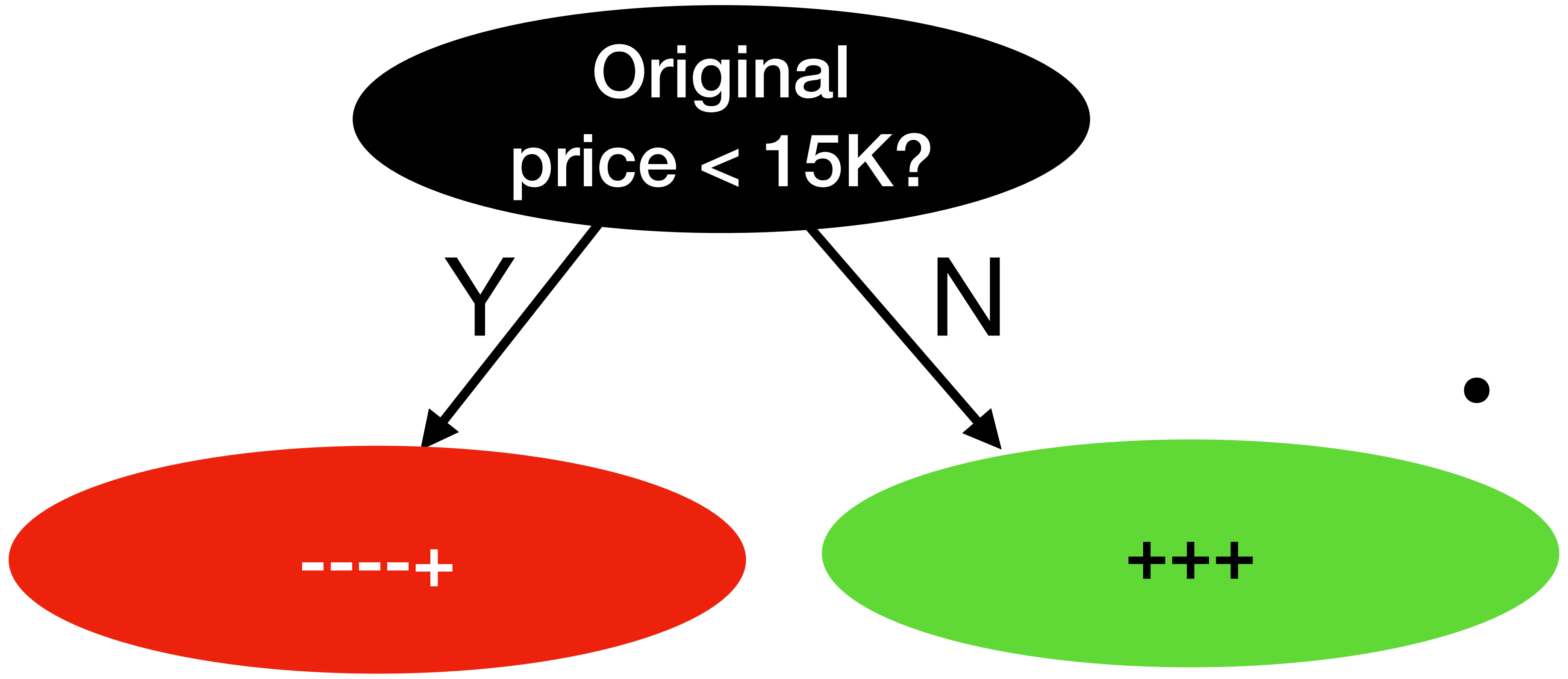
Entropy 0.81

-3/4 lg 3/4 - 1/4 lg 1/4

Expected Entropy

- Each question makes two piles of examples - yes to question, no to question - and calculates entropy for each pile's labels
- Expected entropy: entropies weighted by the branch's fraction of examples
- We pick the question that yields the best expected entropy

- Yes entropy:
  -1/5 lg 1/5 - 4/5 lg 4/5 = 0.72

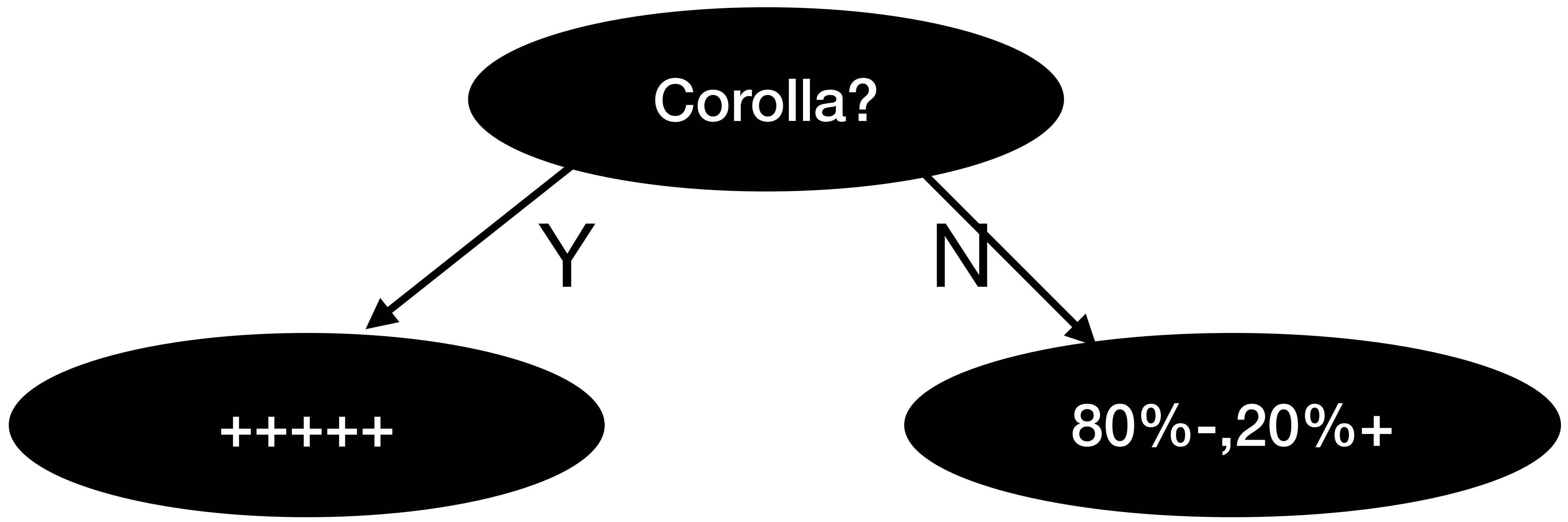


- "No pile" entropy: 0
  (all labels agree)

- Expected entropy: 5/8(0.72) + 3/8(0) = 0.45

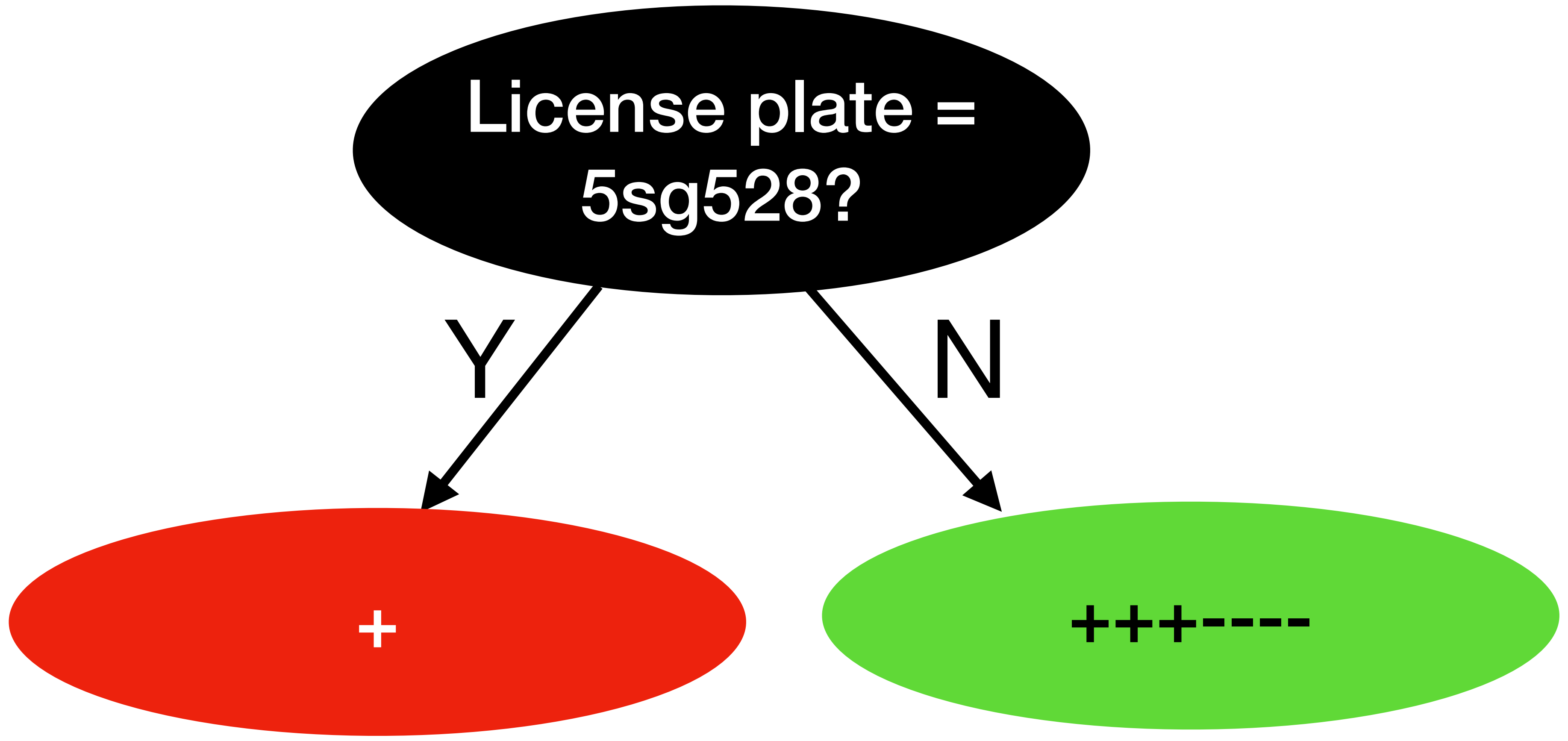
Expected Entropy, example 2

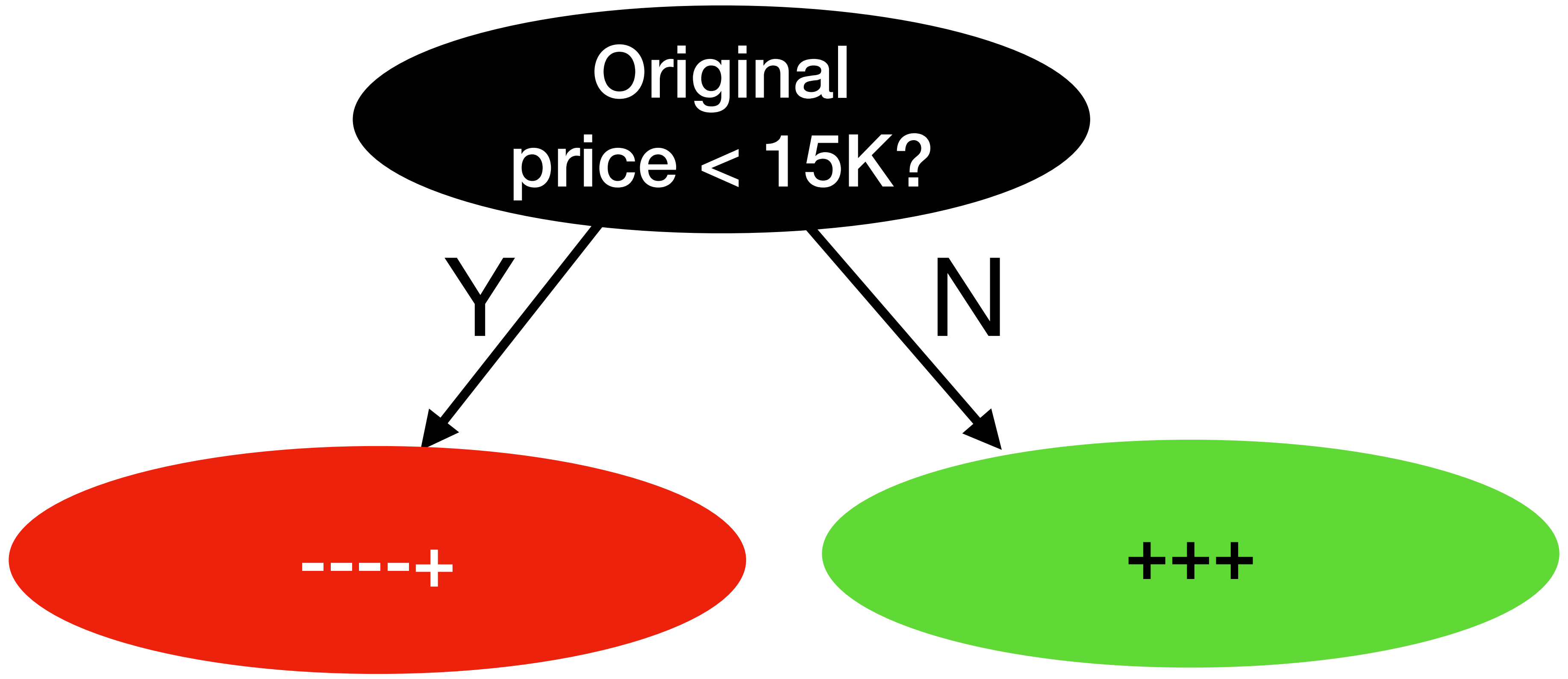
- Question "Is it a Corolla?" affects 5 cars in a 2005 point training set
- All 5 Corollas driven over 200000 miles, so that branch is entropy 0
- 20% of the remaining cars were driven over 200,000 miles, so entropy of that branch is $-0.2 \log_2 0.2 - 0.8 \log_2 0.8 = 0.72$
- Expected entropy is then 5/2005*0 + 2000/2005*0.72 = 0.72



Why Expected Entropy?

- A question that just helps 1 example be classified correctly is not that useful
- Expected entropy rewards questions that help classify many examples





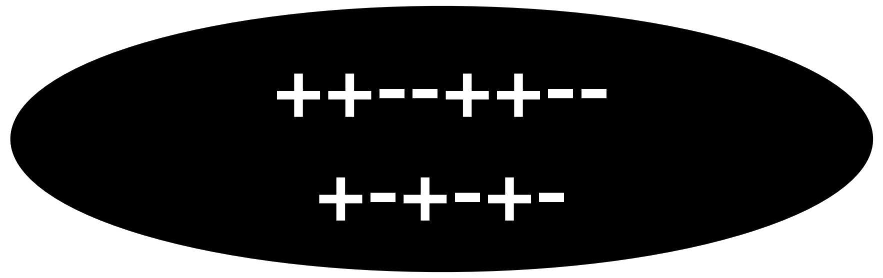
- Exp. entropy: 1/8(0) + 7/8(0.98) = 0.86
- Exp. entropy: 5/8(0.72) + 3/8(0) = 0.45

Decision stumps

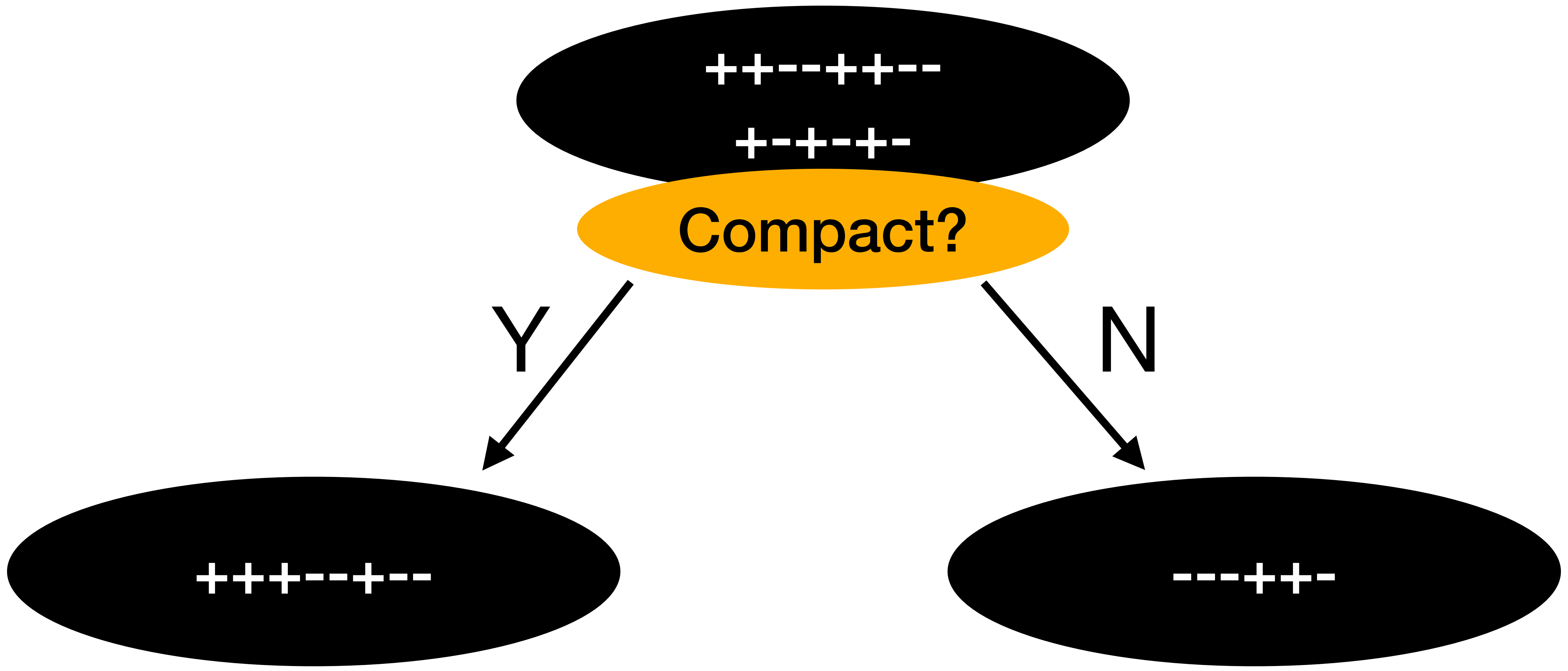
Compact	Hybrid	Price	Class
No	No	9K	-
No	No	8K	-
No	No	7K	-
Yes	No	6K	-
No	No	10K	+
Yes	No	15K	+
No	Yes	17K	+
No	Yes	20K	+

- If we don't recur, we can just pick the single best question according to its expected entropy
- The classifications at the leaves are the "majority vote" of the training examples that land there

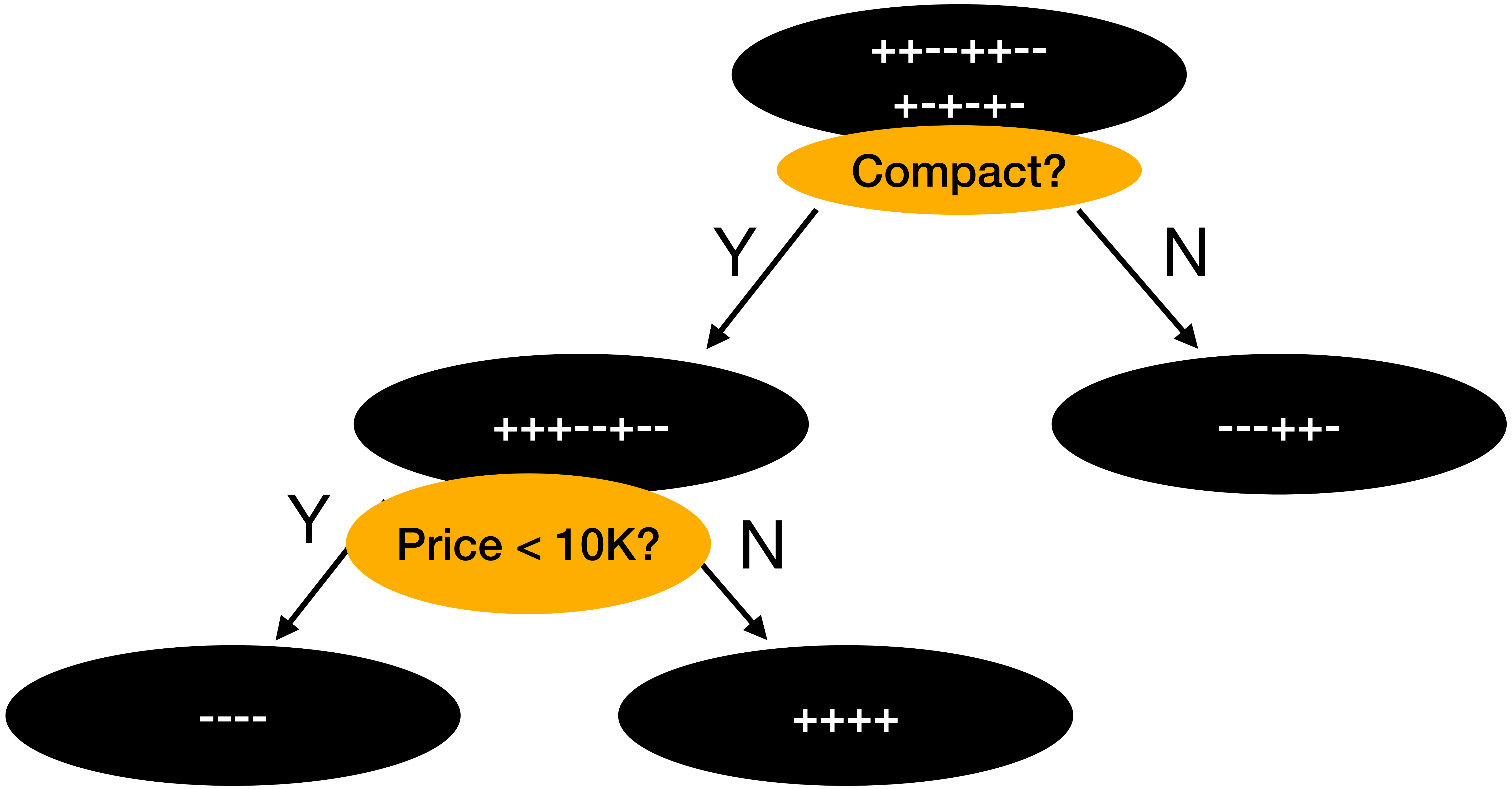
Decision Trees recursively split the training data to minimize entropy



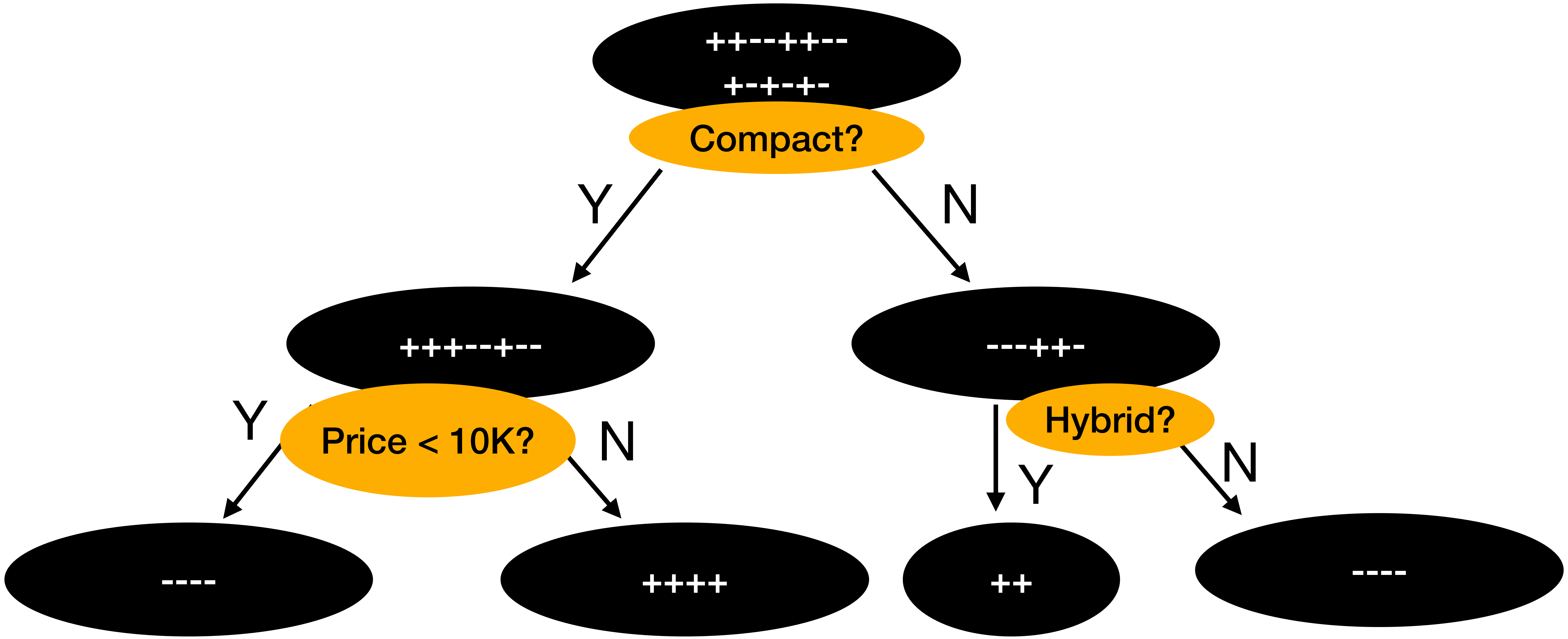
Decision Trees recursively split the training data to minimize entropy



Decision Trees recursively split the training data to minimize entropy



Decision Trees recursively split the training data to minimize entropy



Pseudocode

- DecisionTreeNode(examples):
  - If the examples all agree on a label, return a leaf with that label
  - Iterate through all questions about features to get question Q with best expected entropy
  - If the expected entropy for Q isn't an improvement over the current entropy, stop and make this a leaf with classification according to majority rule
  - Recursively create a "yes" branch with examples that answer "yes" to Q
  - Recursively create a "no" branch with examples that answer "no" to Q

Small sample run: Data and features

Underwater	Legs	Octopus?
No	8	-
No	8	-
No	4	-
Yes	0	-
Yes	8	+
Yes	8	+

Possible features to create questions about:

- Underwater?
- Legs < 0?
- Legs < 4?
- Legs < 8?

Small sample run - Choosing the first feature

Underwater	Legs	Octopus?
No	8	-
No	8	-
No	4	-
Yes	0	-
Yes	8	+
Yes	8	+

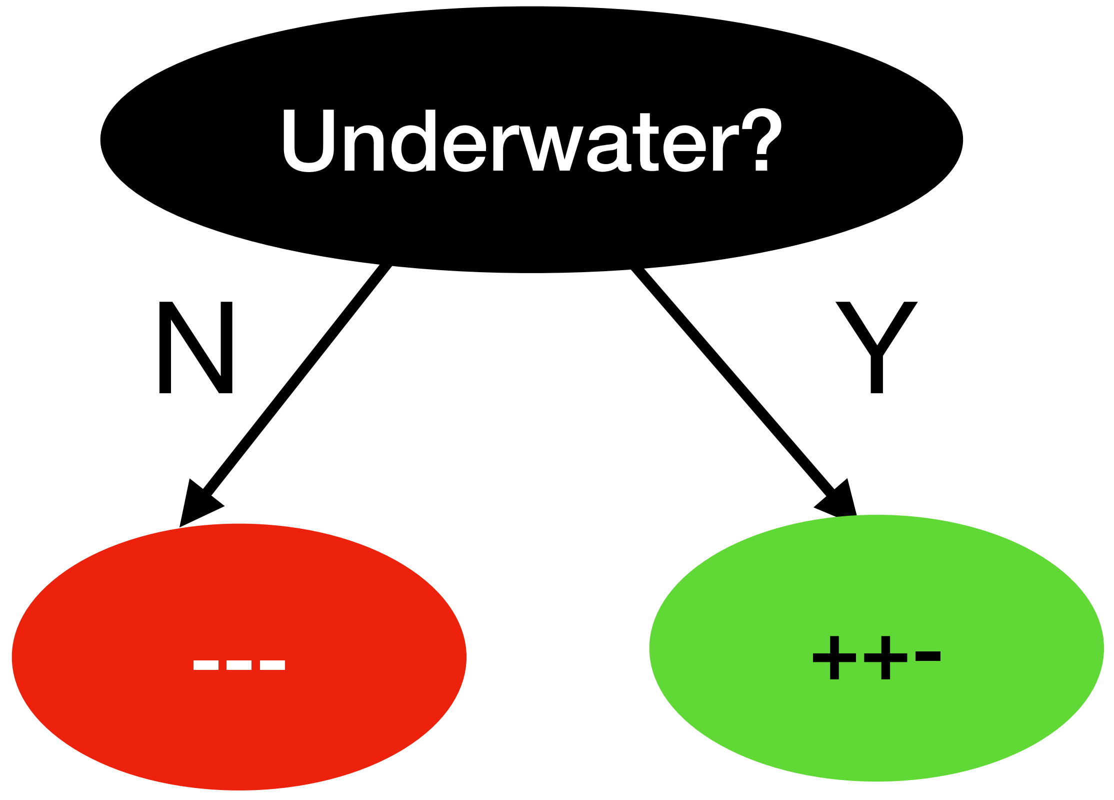
Underwater split:
Answer no: --- (entropy 0)
Answer yes: -++
(entropy $-1/3 \log_2 1/3 - 2/3 \log_2 2/3 = 0.918$)
Expected entropy:
$3/6 * 0 + 3/6 * 0.918 = 0.459$

Best legs split, legs < 8
Answer no: --++ (entropy 1)
Answer yes: -- (entropy 0)
Expected entropy:
$4/6*1 + 2/6*0 = 2/3 = 0.667$

We choose "Underwater?" as the first question.

Small sample run - Choosing the 2nd feature

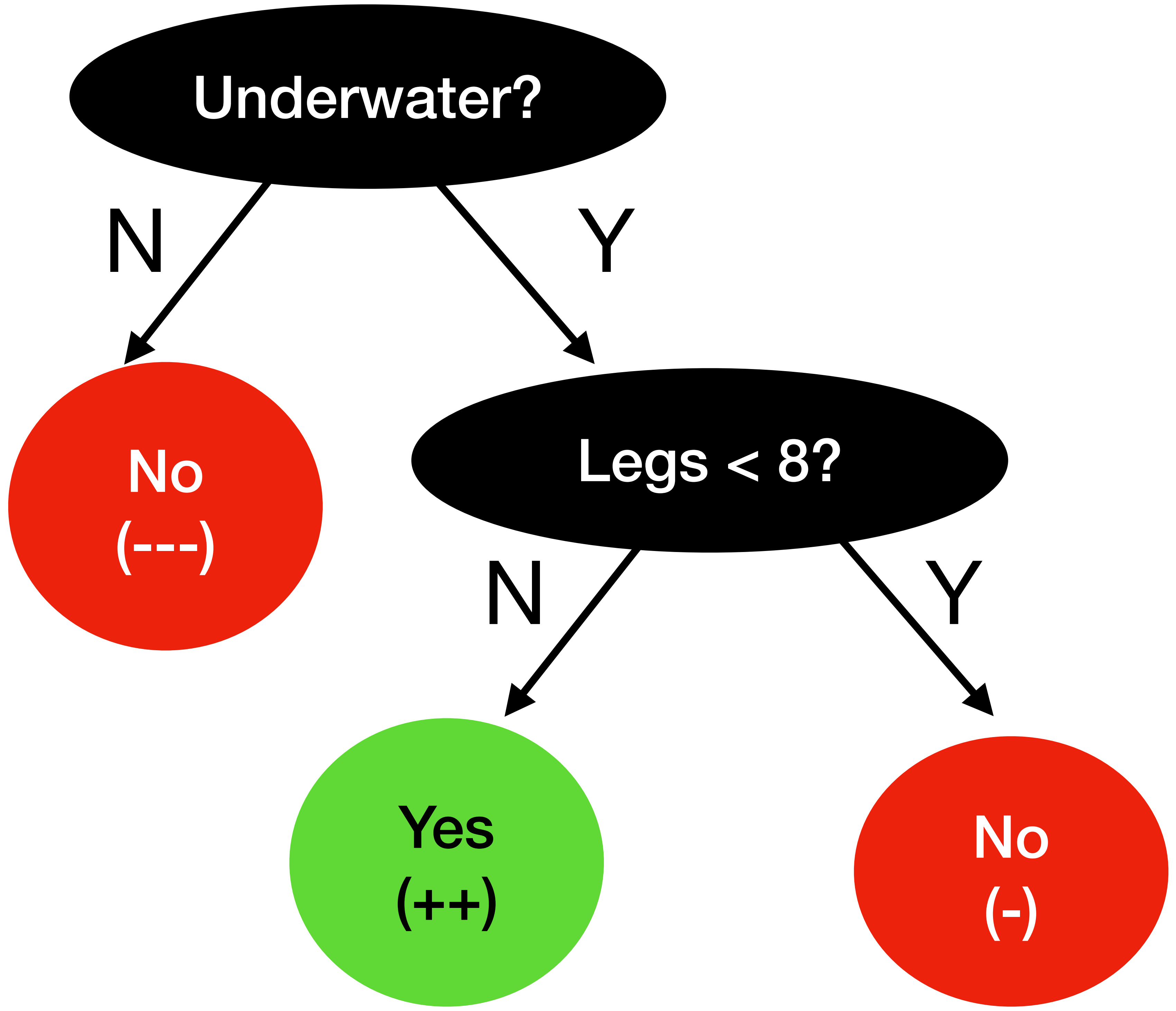
Underwater	Legs	Octopus?
No	8	-
No	8	-
No	4	-
Yes	0	-
Yes	8	+
Yes	8	+



- Nothing more to do for the branch where all agree;
  set this to be a leaf where tree returns "No"
- For remaining branch, try the different legs features that made it to this side:
  legs < 0 has nothing on one side, entropy remains 0.459
  legs < 8 gets entropy 0 on both sides of the split (++, -), expected entropy 0

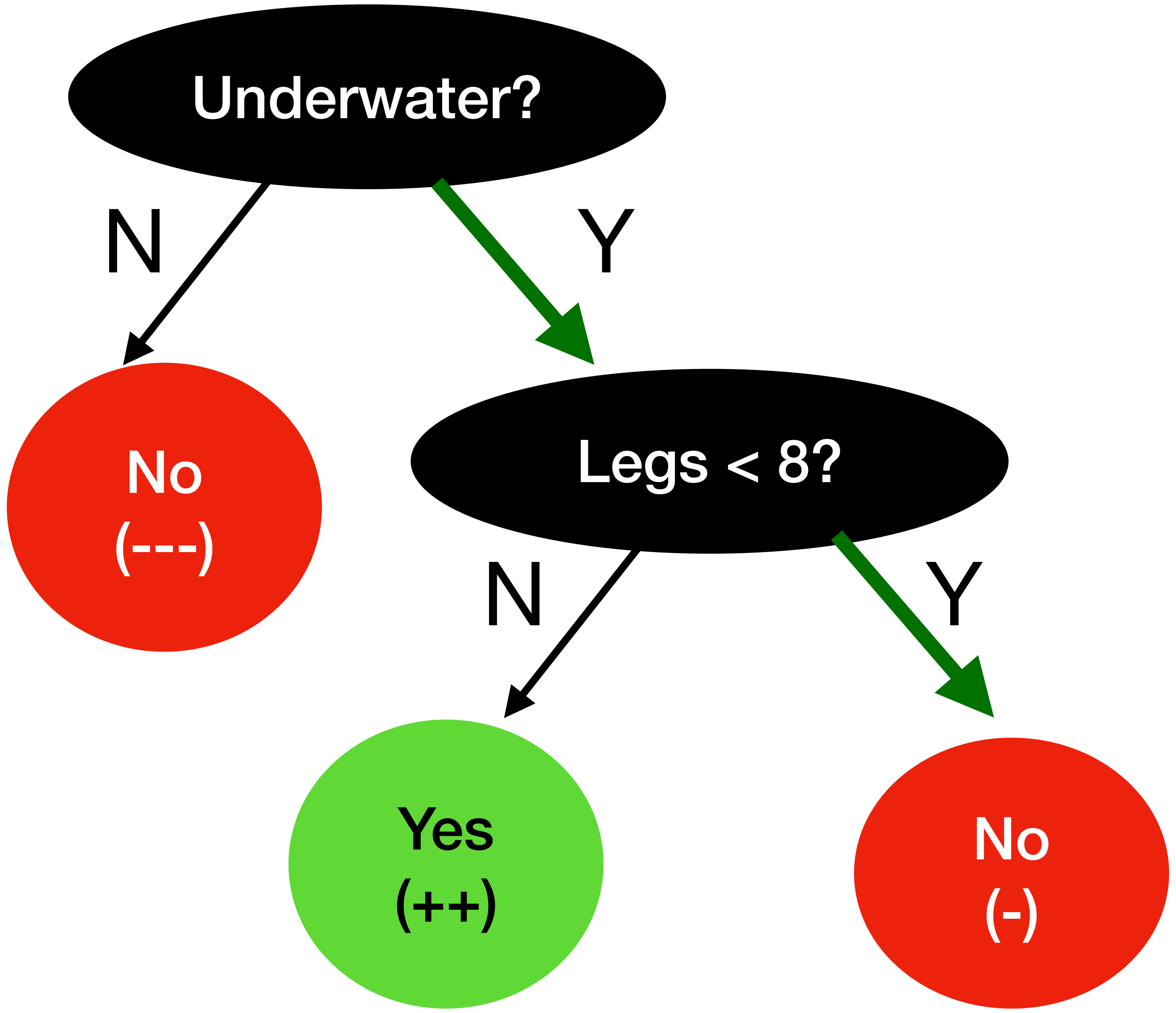
Sample run - the completed octopus-classifying tree

Underwater	Legs	Octopus?
No	8	-
No	8	-
No	4	-
Yes	0	-
Yes	8	+
Yes	8	+



- The tree is now ready to classify new instances it hasn't seen before

Sample run - the completed octopus-classifying tree



- The tree is now ready to classify new instances it hasn't seen before

What if the data labels are inconsistent?

- The data doesn't necessarily lend itself to perfect classification in this way - the exact same features may be labeled differently for different examples.
  - If the diving bell spider on the right were in the dataset (8 legs, underwater), there would be no way to distinguish it from the octopus

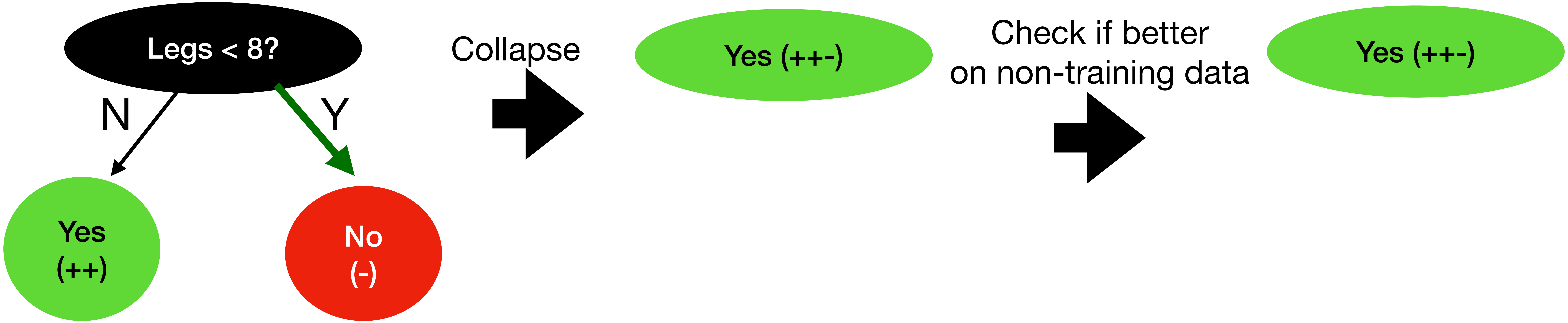
- If the algorithm detects that no feature improves the expected entropy, it stops looking for features and creates a leaf with the majority label of the examples
  - If octopuses are more common than diving bell spiders, the final classification of these will be "octopus"

Overfitting

- It's possible for the learning to create a tree with features that don't generalize well from the training set to unseen examples.
    - Names ("Don't give anyone a loan whose name is Simon!")
    - License plates ("It goes over 200,000 miles if its license plate is 5SG528")
    - Any numerical value that is very specific to an individual, such as an exact height or weight
    - A combination of features that seem harmless in isolation but together uniquely identify an individual ("Blue 2010 Honda Fit garaged in Somerville")
- Overfitting generally results in worse performance on test sets and in deployment

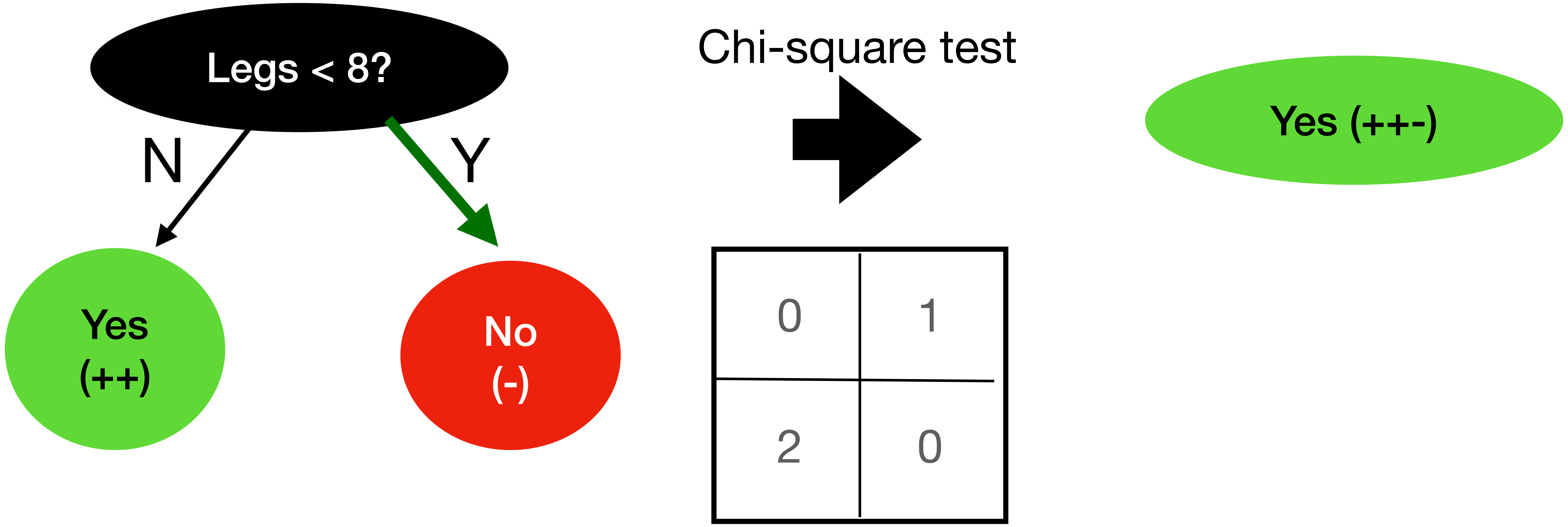
Combatting overfitting

- Various methods exist for **pruning** the decision tree to combat overfitting
  - For each pair of leaves, can check whether tree performs better on a validation set if the leaves are merged back into a single leaf



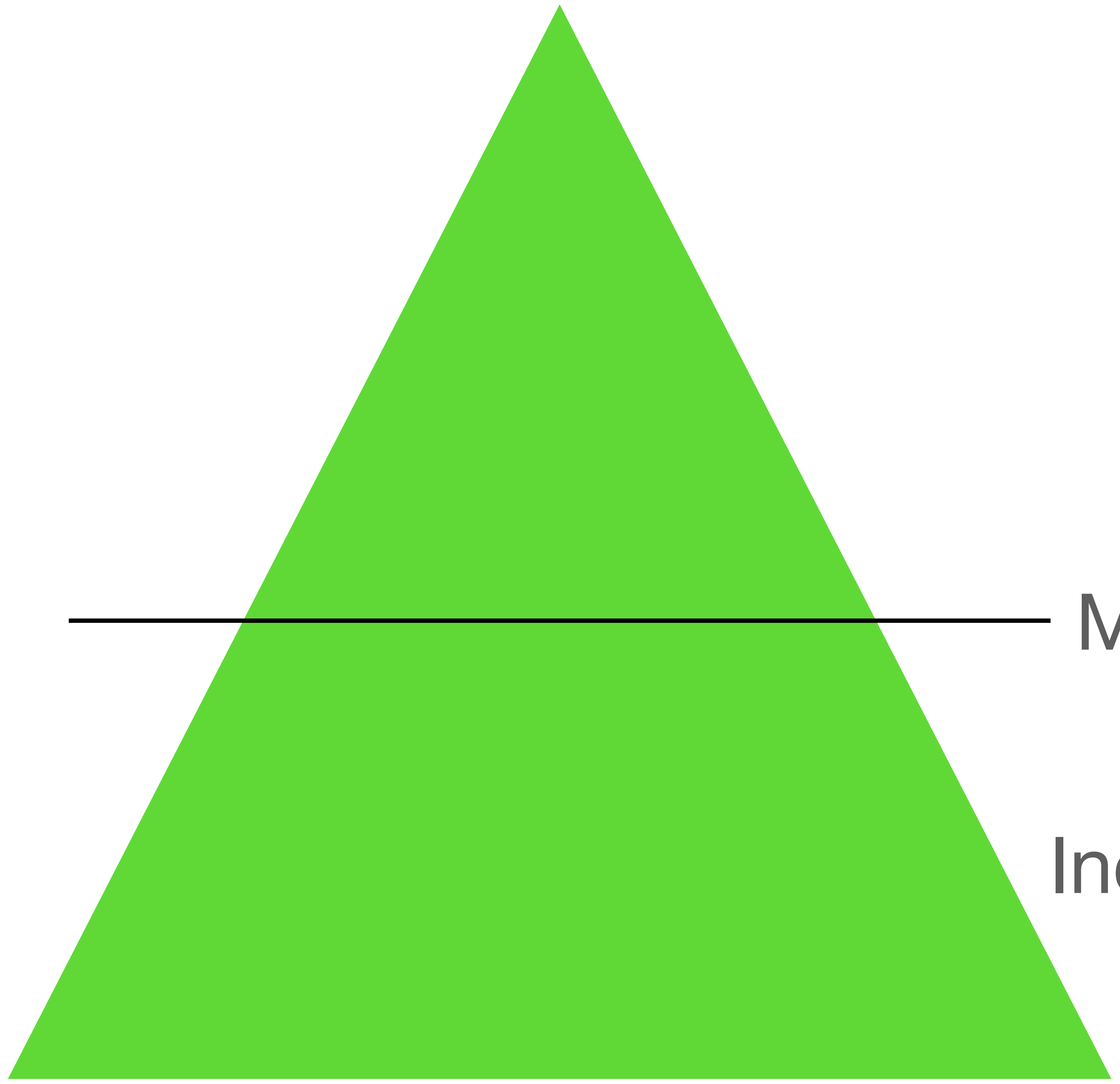
Combatting overfitting

- Various methods exist for **pruning** the decision tree to combat overfitting
  - The leaves could be merged if the relationship between feature and label isn't significant under a chi-square significance test



Combatting overfitting

- Various methods exist for **pruning** the decision tree to combat overfitting
  - A simple method is to just specify the maximum depth of the tree, on the assumption overfitting tends to happen once good features are exhausted



Using Decision Trees in scikit-learn